Prof. dr. Irinel Drăgan

TEHNICI DE BAZĂ ÎN PROGRAMAREA LINIARĂ

Editura tehnică

București — 1976

Scanned with OKEN Scanner

Control ştiinţific: dr. **BERGTHALLER CRISTIAN**
Redactor: **VALENTINA BUCUR**
Tehnoredactor: **VALERIU MORĂRESCU**
Coperta şi supracoperta: **VALENTIN VIŞAN**

Bun de tipar: 15.III.1976. Coli de tipar: 12,75.
Tiraj: 4100+70 exemplare legate. C.Z. 681.142.2.

c. 2542 I. P. INFORMAŢIA,
str. Brezoianu nr. 23–25,
Bucureşti

Scanned with OKEN Scanner

CUVÎNT ÎNAINTE

În procesul de integrare a învățămîntului cu cercetarea și producția, cale de dezvoltare a științei trasată cu claritate de Documentele Congresului al XI-lea al Partidului Comunist Român, un loc important revine activității de modelare a fenomenelor economice, tehnice și sociale. Lucrarea aduce o contribuție importantă la răspîndirea tehnicilor de rezolvare a unor probleme de optimizare din programarea liniară, care este chemată să-și aducă aportul deplin la măreața operă de construire a societății socialiste multilateral dezvoltate în patria noastră.

Lucrarea de față nu este un tratat de programare liniară și nici un curs asupra acestui domeniu, ci o lucrare unde cititorul poate găsi numai aspectele matematice de bază necesare pentru înțelegerea cîtorva tehnici de programare liniară, devenite astăzi de uz curent în mai toate activitățile tehnice și economice. Aparatul matematic, concepte și demonstrații, a fost redus la minimul necesar pentru continuitatea logică a expunerii. Autorul speră că un cititor care a parcurs lucrarea și posedă cunoștințe reduse asupra limbajului FORTRAN poate citi cu ușurință un program aflat în biblioteca unui calculator și, în consecință, îl va putea folosi pentru a rezolva o problemă concretă de programare liniară din domeniul său de activitate. Desigur, este vorba de un program pentru una dintre tehnicile expuse, cititorul care se interesează de rezolvarea unei probleme aparținînd unei anumite clase de probleme de programare liniară trebuind să caute explicații într-o altă lucrare.

Tehnicile puse la îndemîna cititorului în lucrarea de față sînt următoarele: metoda eliminării complete pentru rezolvarea sistemelor liniare (§ 1.2); algoritm pentru inversarea unei matrice pătrate (§ 1.4); metoda simplex primală pentru rezolvarea problemei generale a programării liniare, sub forma obișnuită (§ 3.2, § 3.4.1), și sub forma revizuită (§ 3.5); metoda simplex duală pentru rezolvarea problemei generale a programării liniare, sub forma obișnuită (§ 3.3, § 3.4.2) și sub forma revizuită (§ 3.5); algoritmul simplex primal adaptat pentru rezolvarea programelor liniare parametrice cu un [illegible] care intră liniar în funcția obiectiv (§ 4.2, § 4.3); algoritmul simplex dual adaptat pentru rezolvarea programelor liniare parametrice cu un parametru care intră liniar în termenii liberi (§ 4.4, § 4.5); algoritmii de secționare ai lui R. Gomory pentru rezolvarea problemelor de programare liniară în numere întregi (§ 6.3); algoritmii de ramifi-

Scanned with OKEN Scanner

care și mărginire ai lui B. Korte pentru rezolvarea problemelor de programare liniară în numere întregi (§ 6.4); algoritmul primal-dual al lui L. R. Ford și D. R. Fulkerson pentru rezolvarea problemelor de transport (§ 7.3).

Cum cititorul poate fi interesat numai de o anumită tehnică, pentru care nu este necesară parcurgerea întregii lucrări, autorul consideră util să precizeze că după primele trei capitole poate fi abordat direct oricare dintre ultimele patru. La o primă lectură § 1.4 și § 3.5 pot fi omise, deși acestea sînt esențiale în lucrul cu calculatorul. Cititorul interesat în tehnica dată în § 4.2 și § 4.3 poate omite § 3.3 și § 3.4.2, iar cel interesat în tehnica dată în § 4.4 și § 4.5 poate omite § 3.2 și § 3.4.1. Apoi tehnica dată în § 6.4 poate fi abordată direct, după § 2.1, § 2.2 și § 4.2, iar tehnicile din ultimul capitol, după § 2.1.3.

În literatura consacrată programării liniare există astăzi foarte multe lucrări. Bibliografia dată la sfîrșitul lucrării nu epuizează nici literatura românească a domeniului; comentariul asupra modului de alcătuire a bibliografiei încearcă să explice cum a procedat autorul.

Cititorul avizat va fi surprins de cîte probleme pe care s-a obișnuit a le privi ca esențiale în programarea liniară pot totuși lipsi dintr-o expunere coerentă a unora dintre tehnicile fundamentale. Sperăm că el va fi de acord cu punctul de vedere al autorului, care a urmărit numai aspectele utilitare ale programării liniare. Desigur, au rămas în afara lucrării capitole ale domeniului care au mare valoare practică; credem că prin omiterea lor am contribuit la creșterea bunăvoinței cititorului, în general descurajat de o lucrare voluminoasă, cu prea multe amănunte tehnice.

Este pentru autor cea mai distinsă împrejurare de a-și exprima recunoștința față de regretatul academician Grigore C. Moisil, unul dintre promotorii programării liniare în România.

AUTORUL

CUPRINS

Cuvînt înainte 5

Partea întîi

TEHNICI DE REZOLVARE A PROBLEMEI GENERALE A PROGRAMĂRII LINIARE

1. Sisteme liniare, transformări elementare, explicitarea unui sistem liniar, soluţii de bază 9

1.1. Transformări elementare. 9
1.2. Explicitarea unui sistem liniar, metoda eliminării complete 13
1.3. Pivotajul, matricea de pivotaj. 17
1.4. Inversarea unei matrice. 22
1.5. Soluţii de bază ale unui sistem liniar. 28

2. Problema generală a programării liniare : formularea problemei, explicitarea unui program liniar, pivotaj primal şi dual 32

2.1. Trei modele. 32
2.2. Formularea problemei generale a programării liniare 37
2.3. Programe liniare echivalente, explicitarea unui program liniar. . . . 40
2.4. Soluţii de bază ale unui program liniar. 44
2.5. Pivotaj primal asupra unui program liniar. 47
2.6. Pivotaj dual asupra unui program liniar. 53

3. Metode generale de rezolvare a programelor liniare : metoda simplex primală şi metoda simplex duală 60

3.1. Teoreme fundamentale. 60
3.2. Algoritmul simplex primal pentru rezolvarea programelor liniare aflate sub o formă explicită primal admisibilă 67
3.3. Algoritmul simplex dual pentru rezolvarea programelor liniare aflate sub o formă explicită dual admisibilă 70
3.4. Rezolvarea unui program liniar sub forma standard, prin metoda simplex primală sau duală 73
3.5. Formele revizuite ale algoritmilor simpleciali. 89
3.6. Cazul de degenerare. 93

Partea a doua

TEHNICI DE REZOLVARE PENTRU PROBLEME CONEXE

4. Programe parametrice 96

4.1. Generalităţi 96
4.2. Programe liniare de formă explicită primal admisibilă cu o nedeterminată în funcţia obiectiv 98
4.3. Programe liniare parametrice cu parametrul în funcţia obiectiv. 102
4.4. Programe liniare de formă explicită dual admisibilă cu o nedeterminată în termenii liberi 111
4.5. Programe liniare parametrice cu parametrul în termenii liberi 115

Scanned with OKEN Scanner

5. Teoria dualității programelor liniare. 126
5.1. Probleme duale simetrice. 126
5.2. Probleme duale nesimetrice. 134
5.3. Diagrama lui A. W. Tucker. 138
5.4. Legătura dintre algoritmul simplex primal și algoritmul simplex dual 142

6. Programe liniare în numere întregi. 147
6.1. O altă formă de pivotaj. 147
6.2. Problema programării liniare în numere întregi: cazul total întreg și cazul parțial întreg 155
6.3. Algoritmi de secționare. 159
6.4. Algoritmi de ramificare și mărginire. 169

7. Problema transporturilor 183
7.1. Rețea de transport, flux, secțiune, problema fluxului maxim, problema secțiunii minime 183
7.2. Algoritmul lui L. R. Ford și D. R. Fulkerson pentru rezolvarea fluxului maxim și a problemei secțiunii minime 190
7.3. Algoritmul primal-dual pentru rezolvarea problemei transporturilor. 197

Comentariu la bibliografie.. 202

Bibliografie. 203

Scanned with OKEN Scanner

Partea întîi

TEHNICI DE REZOLVARE A PROBLEMEI GENERALE A PROGRAMĂRII LINIARE

1

SISTEME LINIARE: TRANSFORMĂRI ELEMENTARE, EXPLICITAREA UNUI SISTEM LINIAR, SOLUȚII DE BAZĂ

1.1. TRANSFORMĂRI ELEMENTARE

Se numește *sistem liniar* un sistem de ecuații de forma

$$\left.\begin{aligned} a_1^1 x_1 + a_1^2 x_2 + \ldots + a_1^n x_n &= b_1 \\ a_2^1 x_1 + a_2^2 x_2 + \ldots + a_2^n x_n &= b_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots & \\ a_m^1 x_1 + a_m^2 x_2 + \ldots + a_m^n x_n &= b_m \end{aligned}\right\} \tag{1.1}$$

unde a_i^k și b_i, $(i = 1, \ldots, m;\ k = 1, \ldots, n)$, sînt numere reale. Unui sistem liniar i se asociază: matricea sistemului, formată cu coeficienții necunoscutelor, anume $A = (a_i^k)$, vectorul termenilor liberi $b = (b_i)$ și matricea lărgită a sistemului, anume $(A\ b) = (a_i^k b_i)$; dimensiunile acestor matrice sînt respectiv $m \times n$, $m \times 1$, $m \times (n + 1)$. Dacă se introduce și vectorul n-dimensional $X = (x_k)$, se poate scrie sistemul (1.1) sub forma matriceală

$$AX = b. \tag{1.2}$$

Considerațiile ce urmează se bazează pe faptul evident că un sistem liniar determină în mod unic matricea sa lărgită și reciproc; matricea lărgită a sistemului, uneori o parte a sa, este ansamblul de date care reprezintă sistemul într-un calculator numeric.

Două *sisteme liniare* se numesc *echivalente* dacă orice soluție a primului sistem este soluție și a celui de-al doilea și reciproc.

Scanned with OKEN Scanner

Se numeşte *transformare elementară* asupra unui sistem liniar orice transformare realizată prin una dintre următoarele trei operaţii:

(T_1) înmulţirea unei ecuaţii cu un număr diferit de zero;

(T_2) schimbarea ordinii a două ecuaţii;

(T_3) înmulţirea unei ecuaţii cu un număr diferit de zero, adunarea membru cu membru a rezultatului la o altă ecuaţie şi înlocuirea ecuaţiei la care s-a adunat prin ecuaţia rezultat.

Fiecare dintre operaţiile (T_1), (T_2), (T_3) determină o operaţie corespunzătoare asupra matricei lărgite a sistemului; aceste operaţii sînt: înmulţirea unei linii cu un număr diferit de zero, schimbarea între ele a două linii şi înmulţirea unei linii cu un număr diferit de zero urmată de adunarea rezultatului la o altă linie şi înlocuirea liniei la care s-a adunat prin linia rezultat. Se va numi *transformare elementară asupra matricei lărgite* a unui sistem liniar orice transformare realizată prin una dintre cele trei operaţii menţionate. Observăm că transformările elementare se realizează numeric cu uşurinţă.

Acum, este evident următorul rezultat:

Teorema 1.1. *Orice sistem liniar este transformat într-un sistem liniar echivalent lui prin orice succesiune de transformări elementare.*

Rezultă că printr-o succesiune oarecare de transformări elementare asupra matricei lărgite a unui sistem liniar se obţine matricea lărgită a unui sistem liniar echivalent cu el.

Acest ultim rezultat va fi folosit pe tot parcursul teoriei programelor liniare; de aceea, cele trei operaţii asupra matricei lărgite a unui sistem liniar vor fi întîlnite în cea mai mare parte a exemplelor ce urmează. Pentru comoditatea limbajului vom denumi transformare (T_h) asupra matricei lărgite a unui sistem liniar, transformarea corespunzătoare operaţiei (T_h) asupra sistemului, ($h = 1, 2, 3$). Avertizăm cititorul că transformările de acelaşi tip, dar realizate asupra coloanelor matricei lărgite a unui sistem liniar, sînt interzise, deoarece prin asemenea transformări nu s-ar obţine un sistem liniar echivalent cu cel dat.

Să arătăm că o transformare (T_h) este de fapt o înmulţire între două matrice, dintre care una este matricea lărgită a sistemului.

Scanned with OKEN Scanner

Pentru aceasta introducem patru tipuri de matrice pătrate de ordinul m :

$$M_1 = \begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & \alpha & \\ & & & & \ddots \\ & & & & & 1 \end{pmatrix} \begin{matrix} \\ \\ \\ \to i \\ \\ \\ \end{matrix} \tag{1.3}$$

care diferă de matricea unitate prin elementul din poziţia (i, i) înlocuit cu numărul α, diferit de zero ;

$$M_2 = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 0 & \dots & 1 & \\ & & \vdots & & \vdots & \\ & & 1 & \dots & 0 & \\ & & & & & \ddots \\ & & & & & & 1 \end{pmatrix} \begin{matrix} \\ \\ \to i \\ \\ \to j \\ \\ \\ \end{matrix} \tag{1.4}$$

care diferă de matricea unitate prin elementele din poziţia (i, i), (j, j), înlocuite cu zero, şi cele din poziţiile (i, j), (j, i), înlocuite cu unu ;

$$M_3' = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & \dots & \alpha & \\ & & & \ddots & & \\ & & & & 1 & \\ & & & & & \ddots \\ & & & & & & 1 \end{pmatrix} \begin{matrix} \\ \\ \to i \\ \\ \to j \\ \\ \\ \end{matrix} \tag{1.5}$$

care diferă de matricea unitate prin elementul din poziţia (i, j), înlocuit cu numărul α, diferit de zero;

$$M_3'' = \begin{pmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & \vdots & \ddots & & \\ & & \alpha & \ldots & 1 & \\ & & & & & \ddots \\ & & & & & & 1 \end{pmatrix} \begin{matrix} \\ \\ \to i \\ \\ \to j \\ \\ \\ \end{matrix} \tag{1.6}$$

care diferă de matricea unitate prin elementul din poziţia (j, i), înlocuit cu numărul α, diferit de zero.

O matrice avînd una din formele M_1, M_2, M_3', M_3'' va fi numită *matrice elementară*.

Teorema 1.2. *O matrice obţinută prin înmulţirea la stînga a matricei lărgite a unui sistem liniar cu o matrice elementară este matricea lărgită a unui sistem liniar echivalent celui considerat.*

Într-adevăr, dacă $\mathscr{A} = (Ab)$ este matricea lărgită a unui sistem liniar şi calculăm produsele $M_1\mathscr{A}$, $M_2\mathscr{A}$, $M_3'\mathscr{A}$, $M_3''\mathscr{A}$, observăm că aceste produse realizează transformările (T_h), $(h = 1, 2, 3)$. Atunci, rezultatul de mai sus decurge din teorema 1.1.

Anticipăm afirmînd că teorema 1.2 va fi folosită pentru justificarea formelor revizuite ale algoritmilor simpleciali de rezolvare a problemelor de programare liniară (v. § 3.5).

Exemplul 1.1. Sistemul liniar

$$\left.\begin{aligned} x_1 - x_2 + 2x_3 - x_4 &= 2 \\ 2x_1 + x_2 + x_4 &= 1 \\ x_2 + 3x_3 &= -1 \end{aligned}\right\} \tag{1.7}$$

are matricea lărgită

$$\mathscr{A} = \left(\begin{array}{cccc|c} 1 & -1 & 2 & -1 & 2 \\ 2 & 1 & 0 & 1 & 1 \\ 0 & 1 & 3 & 0 & -1 \end{array}\right) \tag{1.8}$$

în care linia punctată verticală separă matricea sistemului de vectorul termenilor liberi.

Matricele elementare

$$M_1 = \begin{pmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad M_2 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix},$$

$$M_3' = \begin{pmatrix} 1 & -2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \qquad M_3'' = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \tag{1.9}$$

realizează prin înmulțire cu $\mathscr{A}$ următoarele transformări : $M_1\mathscr{A}$ se obține prin înmulțirea cu 2 a liniei întîi din $\mathscr{A}$; $M_2\mathscr{A}$ se obține prin schimbarea între ele a liniei întîi cu linia a treia în $\mathscr{A}$; $M_3'\mathscr{A}$ se obține prin înmulțirea cu -2 a liniei a doua din $\mathscr{A}$ urmată de adunarea la linia întîi ; $M_3''\mathscr{A}$ se obține prin înmulțirea cu -1 a liniei întîi din $\mathscr{A}$ urmată de adunarea la linia a treia. Toate aceste afirmații se pot verifica dacă se calculează produsele respective.

1.2. EXPLICITAREA UNUI SISTEM LINIAR, METODA ELIMINĂRII COMPLETE

Vom spune că un *sistem liniar* este *explicitat* dacă matricea sistemului conține toate coloanele matricei unitate de ordin egal cu numărul ecuațiilor sistemului.

Desigur este posibil ca aceste coloane să fie dispuse astfel încît matricea unitate să formeze un bloc al matricei sistemului ; remarcăm însă că definiția formulată nu cere numaidecît prezența matricei unitate, ci numai a coloanelor sale în matricea sistemului. Remarcăm de asemenea că într-un sistem explicitat numărul variabilelor este cel puțin egal cu numărul ecuațiilor.

Într-un sistem liniar explicitat sînt două tipuri de variabile : unele, ai căror coeficienți formează coloanele matricei unitate, altele, ai căror coeficienți formează celelalte coloane ale matricei sistemului. Ca de obicei, primele vor fi denumite *variabile principale*, iar celelalte *variabile secundare* ; în teoria programării liniare variabilele principale mai sînt denumite bazice, iar variabilele secundare, nebazice. Se înțelege, aceste denumiri arată anumite legături cu conceptul de „bază" din algebra liniară ; ele s-au încetățenit în teoria programelor liniare mai ales datorită modului de expunere a teoriei adoptat de autorii primelor manuale consacrate domeniului [1, 2]. Renunțînd la maniera de expunere din lucrările menționate,

în interesul prezentării cît mai elementare a teoriei, vom renunța și la termenii de variabilă bazică și variabilă nebazică.

Într-un sistem liniar explicitat, fie I mulțimea indicilor variabilelor principale și K mulțimea indicilor variabilelor secundare. Vom putea spune acum, mai precis, că sistemul liniar este explicitat în raport cu mulțimea de variabile x_i, $(i \in I)$. Într-adevăr, nu ne rămîne decît să trecem în membrul drept toți termenii conținînd variabile secundare și vom obține variabilele principale ca funcții de variabilele secundare.

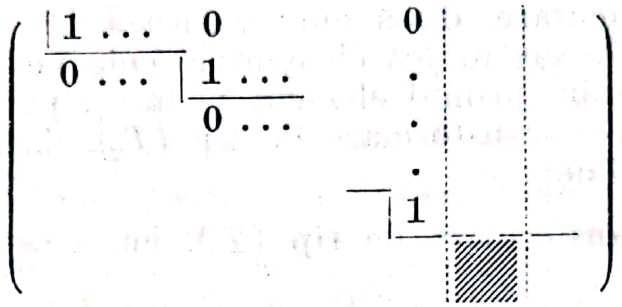
Se știe din teoria sistemelor liniare că a rezolva un sistem liniar înseamnă a găsi o parte dintre variabilele sistemului ca funcții de celelalte, astfel încît substituind aceste funcții în sistem, în locul variabilelor respective, să se obțină identități. Atunci este clar că a explicita un sistem liniar înseamnă a rezolva acel sistem și reciproc. În cele ce urmează, preferăm denumirea de sistem explicitat celei de sistem rezolvat, întrucît vom introduce conceptul asemănător de program liniar explicitat, care nu înseamnă deloc program liniar rezolvat.

Este bine cunoscut că un sistem liniar poate fi compatibil sau incompatibil, iar în primul caz el poate fi explicitat. Ne propunem să arătăm cum se poate stabili dacă un sistem liniar este compatibil sau incompatibil, iar în primul caz cum putem să-l și explicităm, folosindu-ne numai de transformări elementare. Metoda de explicitare pe care o vom formula va fi denumită *metoda eliminării complete*. Subliniem că această metodă este convenabilă din punct de vedere numeric.

Ideea centrală a metodei constă în determinarea unui șir finit de sisteme liniare echivalente celui dat inițial, astfel încît matricea lărgită a oricărui sistem din șir să conțină o coloană a matricei unitate în plus față de matricea sistemului precedent. Se înțelege, procedeul păstrează la fiecare pas toate coloanele deja construite ale matricei unitate și va fi oprit atunci cînd va fi imposibil să se construiască o nouă coloană.

Să presupunem că aplicînd metoda eliminării complete sistemului liniar dat (S^0) am obținut succesiv sistemele liniare (S^1), (S^2), ..., (S^r), echivalente cu (S^0) și ale căror matrice A^1, A^2, ..., A^r conțin respectiv o coloană a matricei unitate, două coloane ale matricei unitate, ..., r coloane ale matricei unitate, de același ordin cu numărul ecuațiilor sistemului (S^0). Să admitem, ceea ce nu restrînge generalitatea, că am obținut succesiv primele r coloane ale matricei

unitate şi că necunoscutele principale deja determinate sînt x_{i_1}, x_{i_2}, ..., x_{i_r}, $(i_1 < i_2 < \ldots < i_r;\ i_1 = 1)$, iar aspectul matricei lărgite a sistemului (S^r) este următorul :



La stînga primei linii punctate verticale, sub linia în scară toate elementele sînt nule, iar deasupra liniei în scară, în coloanele necunoscutelor examinate care nu sînt principale, pot fi elemente oarecare. La dreapta celei de-a doua linii punctate verticale sînt termenii liberi rezultaţi din calcule. Desigur, cele două linii punctate verticale ar putea coincide, adică $i_r = n$, după cum s-ar putea ca ultima treaptă să fie sub matrice, adică, $r = m$. În afară de aceste două cazuri, cînd aplicarea procedeului este terminată, va trebui să construim, în decursul pasului $r + 1$, cea de-a $r + 1$-a coloană din matricea unitate, utilizînd transformări elementare.

Operaţiile pasului $r + 1$ sînt următoarele :

1° Stabilim dacă $r = m$ sau $r < m$; în primul caz sistemul este explicitat şi procedeul este oprit, în caul al doilea se trece mai departe.

2° Stabilim dacă $i_r = n$ sau $i_r < n$, dar în matricea sistemului la dreapta primei linii punctate verticale sub ultima treaptă toate elementele sînt nule, ori $i_r < n$ şi există cel puţin un element nenul (cercetarea o facem coloană cu coloană în spaţiul haşurat) ; în primul caz trecem la operaţia 3°, în cazul al doilea la operaţia 4°.

3° Stabilim dacă toate elementele ultimei coloane situate în liniile $r + 1, \ldots, m$ sînt nule sau există măcar un element nenul ; în ambele cazuri procedeul este oprit : în primul caz sistemul este compatibil, sistemul cu ultima matrice este un sistem explicitat de r ecuaţii, $m - r$ ecuaţii ale sistemului iniţial erau combinaţii ale celorlalte ; în cazul al doilea sistemul dat este incompatibil, căci în sistemul (S^r) echivalent lui există o ecuaţie de forma $0 \cdot x_1 + 0 \cdot x_2 + \ldots + 0 \cdot x_n = \beta$, $(\beta \neq 0)$, care nu poate fi satisfăcută de nici un sistem de valori ale necunoscutelor.

Scanned with OKEN Scanner

4° Să presupunem că în timpul operației 2° s-a găsit un prim element nenul în una dintre liniile $r+1, \ldots, m$ în coloana i_{r+1}, în timp ce în aceleași linii în coloanele precedente toate elementele sînt nule; atunci, vom putea construi sistemul liniar (S^{r+1}) prin transformări elementare, după cum urmează:

a) cercetînd de sus în jos elementele coloanei i_{r+1}, începînd cu linia $r+1$, stabilim primul element nenul γ, pe care-l aducem în linia $r+1$ printr-o transformare de tip (T_2), dacă el nu se găsea deja pe această linie;

b) printr-o transformare de tip (T_1), cu $\alpha = \frac{1}{\gamma}$, realizăm în poziția $(r+1, i_{r+1})$ un element egal cu unu;

c) prin $m-1$ transformări de tip (T_3), în care linia $r+1$ este succesiv înmulțită cu numere convenabile și adunată de fiecare dată la o altă linie, realizăm elemente nule în toate celelalte poziții ale coloanei i_{r+1}. Cu aceasta operația 4° este terminată și trecem la un nou pas.

Este clar că operația 4° determină matricea lărgită a unui sistem liniar (S^{r+1}), echivalent cu (S^0), și realizează o matrice lărgită de aceeași formă cu aceea a sistemului (S^r), însă cu o treaptă în plus. Este evidentă teorema de convergență:

Teorema 1.3. *Fiind dat un sistem liniar* (S^0), *printr-un număr finit de transformări elementare se obține fie rezultatul că* (S^0) *este un sistem incompatibil, fie un sistem explicitat echivalent cu* (S^0).

Remarcăm de asemenea faptul că în urma explicitării unui sistem liniar de m ecuații se poate obține un sistem liniar echivalent de $r-m$ ecuații, ceea ce atestă existența unor perechi de sisteme echivalente cu număr diferit de ecuații.

Exemplul 1.2. Fie sistemul liniar (S^0):

$$\left.\begin{aligned} x_1 - 2x_2 + 2x_3 - x_4 &= 2 \\ 2x_1 + x_2 + x_4 &= 1 \\ 3x_1 + 4x_2 - 2x_3 + 3x_4 &= \lambda \end{aligned}\right\} \tag{1.10}$$

Să discutăm compatibilitatea sistemului pentru diferitele valori ale lui λ și în cazurile de compatibilitatea să găsim un sistem explicitat echivalent cu (S^0). Considerăm matricea lărgită a sistemului

$$\mathscr{A} = \left(\begin{array}{cccc|c} 1 & -2 & 2 & -1 & 2 \\ 2 & 1 & 0 & 1 & 1 \\ 3 & 4 & -2 & 3 & \lambda \end{array}\right). \tag{1.11}$$

Pasul 1. Înmulţim linia întîi cu 1 şi o trecem într-o nouă matrice $\mathscr{A}^1$; înmulţim această linie cu -2, o adunăm la linia a doua din $\mathscr{A}$ şi rezultatul îl scriem în linia a doua din $\mathscr{A}^1$; înmulţim aceeaşi linie cu -3 o adunăm la linia a treia din $\mathscr{A}$ şi rezultatul îl scriem în linia a treia din $\mathscr{A}^1$; obţinem

$$\mathscr{A}^1 = \left(\begin{array}{c|ccc:c} 1 & -2 & 2 & -1 & 2 \\ \hline 0 & 5 & -4 & 3 & -3 \\ 0 & 10 & -8 & 6 & \lambda-6 \end{array}\right), \tag{1.12}$$

matricea lărgită a unui sistem liniar (S^1), echivalent cu (S^0).

Pasul 2. Înmulţim linia a doua cu 1/5 şi rezultatul îl scriem în linia a doua dintr-o nouă matrice $\mathscr{A}^2$; înmulţim această linie cu 2, o adunăm la linia întîi din $\mathscr{A}^1$ şi rezultatul îl scriem în linia întîi din $\mathscr{A}^2$; înmulţim aceeaşi linie cu -10, o adunăm la linia a treia din $\mathscr{A}^1$ şi rezultatul îl scriem în linia a treia din $\mathscr{A}^2$; obţinem

$$\mathscr{A}^2 = \left(\begin{array}{cc|cc:c} 1 & 0 & 2/5 & 1/5 & 4/5 \\ 0 & 1 & -4/5 & 3/5 & -3/5 \\ \hline 0 & 0 & 0 & 0 & \lambda \end{array}\right), \tag{1.13}$$

matricea lărgită a unui sistem liniar (S^2), echivalent cu (S^0).

Pasul 3. Acum $r = 2$, deci $r<m$; toate elementele din linia a treia, dintre cele două linii punctate verticale sînt nule, prin urmare, spre deosebire de paşii anteriori trecem la operaţia 3°; concluzia este:

— dacă $\lambda = 0$, sistemul dat este compatibil, sistemul cu ultima matrice

$$\left.\begin{aligned} x_1 + \frac{2}{5}x_3 + \frac{1}{5}x_4 &= \frac{4}{5} \\ x_2 - \frac{4}{5}x_3 + \frac{3}{5}x_4 &= -\frac{3}{5} \end{aligned}\right\} \tag{1.14}$$

este un sistem liniar explicitat de două ecuaţii, echivalent cu (S^0);

— dacă $\lambda \neq 0$, sistemul dat este incompatibil.

1.3. PIVOTAJUL, MATRICEA DE PIVOTAJ

Metoda eliminării complete pentru explicitarea unui sistem liniar dat (S^0) conţine ca element esenţial operaţia de trecere de la matricea lărgită a unui sistem liniar (S^r), echivalent cu (S^0), la matricea lărgită a unui alt sistem liniar (S^{r+1}), echivalent cu (S^0), prin transformări elementare, (operaţiile 4°b şi 4°c). Această operaţie de trecere, repetată în fiecare pas al metodei, cu excepţia pasului final, rezolvă de fapt următoarea problemă:

Scanned with OKEN Scanner

Se dă o matrice de tip $p \times q$

$$M = \begin{pmatrix} \alpha_1^1 & \dots & \alpha_1^k & \dots & \alpha_1^q \\ \vdots & & \vdots & & \vdots \\ \alpha_i^1 & \dots & \alpha_i^k & \dots & \alpha_i^q \\ \vdots & & \vdots & & \vdots \\ \alpha_p^1 & \dots & \alpha_p^k & \dots & \alpha_p^q \end{pmatrix}, \tag{1.15}$$

în care elementul α_i^k este nenul; se cere ca prin transformări elementare să se realizeze în coloana k din matricea transformată coloana i dintr-o matrice unitate de ordinul p. Cu alte cuvinte, dorim să realizăm 1 în locul elementului α_i^k și 0 în locul celorlalte elemente din coloana k.

Într-adevăr, în pasul $r+1$ al metodei eliminării complete avem de rezolvat exact această problemă pentru matricea lărgită a sistemului liniar (S^r), cu datele particulare $p = m$, $q = n+1$, $i = r+1$, $k = i_{r+1}$.

Problema formulată este însă o subproblemă nu numai pentru problema explicitării unui sistem liniar, ci și pentru aceea a rezolvării unui program liniar. De aceea, cu toate că deja s-a arătat cum aceasta poate fi rezolvată, vom prezenta și un alt mod de rezolvare. Înainte de aceasta, introducem însă unele denumiri provenite din teoria programelor liniare.

Elementul nenul α_i^k dat se va numi *pivot*, iar ansamblul operațiilor de transformare a matricei M într-o matrice cu coloana i din matricea unitate de ordin p în coloana k se va numi *pivotajul cu elementul pivot* α_i^k.

Așadar, un pivotaj cu elementul pivot α_i^k se poate realiza printr-o transformare de tip (T_1) și $p-1$ transformări de tip (T_3), ca în metoda eliminării complete. Remarcăm că la realizarea pivotajului asupra matricei M este necesară prezența în memoria operativă a calculatorului a întregii matrice. Dacă dorim însă să construim numai o anumită coloană h din noua matrice, este suficientă doar prezența coloanelor h și k din M. Rezultă că atunci cînd dorim să realizăm mai multe pivotaje succesiv și să construim numai o parte a matricei finale, sînt suficiente în memoria calculatorului numai coloanele corespunzătoare din matricea inițială și coloanele pivoților succesivi.

Scanned with OKEN Scanner

Observaţiile tocmai prezentate, împreună cu faptul stabilit în § 1.1 că realizarea unei transformări elementare asupra unei matrice este rezultatul înmulţirii la stînga a acelei matrice cu o anumită matrice elementară, ne conduc la un alt procedeu de realizare a unui pivotaj, numeric diferit de cel expus în § 1.2. Pentru a ajunge la formularea acestui procedeu de pivotaj asupra matricei M va trebui să stabilim matricele elementare care înmulţesc succesiv la stînga matricele supuse transformărilor elementare în operaţiile 4°b şi 4°c.

Matricea

$$M^i = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1/\alpha_i^k & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix} \rightarrow i \qquad (1.16)$$

realizează înmulţirea liniei i din M cu factorul $1/\alpha_i^k$ şi, prin urmare, produsul M^iM conţine 1 în locul elementului α_i^k, iar linia i din acest produs este exact linia i din matricea finală. Cu această linie va trebui să lucrăm la transformarea celorlalte linii din M^iM.

Matricele

$$M^1 = \begin{pmatrix} 1 & \cdots & -\alpha_1^k & & \\ & \ddots & \vdots & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix} \rightarrow i$$

$$\cdots\cdots\cdots\cdots \qquad (1.17)$$

$$M^{i-1} = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & -\alpha_{i-1}^k & \\ & & & 1 & \\ & & & & \ddots \\ & & & & & 1 \end{pmatrix} \rightarrow i$$

realizează succesiv înmulţirea liniei i din M^1M cu $-\alpha_1^k$ urmată de adunarea rezultatelor la linia întîi, ..., înmulţirea liniei i din

Scanned with OKEN Scanner

$M^{i-2} \ldots M^1 M^i M$ cu $-\alpha_{i-1}^k$ urmată de adunarea rezultatelor la linia $i-1$, şi, prin urmare, produsul $M^{i-1} \ldots M^1 M^i M$ conţine elemente nule în poziţiile $(1, k) \ldots, (i-1, k)$. Este clar că matricele (1.17) trebuie considerate numai cînd $i > 1$.

Matricele

$$M^{i+1} = \begin{pmatrix} 1 & & & & \\ & \ddots & & & \\ & & 1 & & \\ & & -\alpha_{i+1}^k & \ddots & \\ & & & & 1 \end{pmatrix} \rightarrow i$$

$$\ldots\ldots\ldots\ldots\ldots\ldots \qquad (1.18)$$

$$M^p = \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & \vdots & \ddots \\ & & -\alpha_p^k & \ldots 1 \end{pmatrix} \rightarrow i$$

realizează succesiv: înmulţirea liniei i din $M^{i-1} \ldots M^1 M^i M$ cu $-\alpha_{i+1}^k$ urmată de adunarea rezultatelor la linia $i+1, \ldots,$ înmulţirea liniei i din $M^{p-1} \ldots M^{i+1} M^{i-1} \ldots M^1 M^i M$ cu $-\alpha_p^k$ urmată de adunarea rezultatelor la linia p, şi, prin urmare, produsul

$$N = M^p \ldots M^{i+1} M^{i-1} \ldots M^1 M^i M \qquad (1.19)$$

conţine elemente nule şi în poziţiile $(i+1, k), \ldots, (p, k)$. Este clar că matricele (1.18) trebuie considerate numai cînd $i < p$.

Aşadar, pivotajul cu elementul pivot α_i^k dat asupra matricei M este realizat prin înmulţiri succesive la stînga cu matricele elementare (1.16), (1.17), (1.18), iar matricea nouă rezultată este N. Dacă dorim să construim numai o coloană n^h din N, este suficient să dispunem de matricele elementare şi de coloana m^h din M şi să efectuăm produsul $n^h = M^p \ldots M^{i+1} M^{i-1} \ldots M^1 M^i m^h$.

Scanned with OKEN Scanner

Ceea ce este foarte important pentru realizarea pivotajului este faptul că produsul N se poate efectua şi sub forma

$$N = (M^p \dots M^{i+1}M^{i-1} \dots M^1M^i)\,M, \tag{1.20}$$

produsul din paranteză putînd fi scris direct, fără a se scrie în prealabil factorii săi, plecînd de la coloana k a matricei M.

Într-adevăr, dacă ţinem seamă de ce anume transformări elementare se realizează prin înmulţirea lui M^i la stînga succesiv cu matricele $M^1, \dots, M^{i-1}, M^{i+1}, \dots, M^p$ şi aplicăm aceste transformări matricei M^i, aşa cum se cere la calculul produsului $M^p \dots M^{i+1}M^{i-1} \dots M^1M^i$, se verifică imediat că

$$M^p \dots M^{i+1}M^{i-1} \dots M^1M^i = \begin{pmatrix} 1 & & -\alpha_1^k/\alpha_i^k & & \\ & \ddots & \vdots & & \\ & & -\alpha_{i-1}^k/\alpha_i^k & & \\ & & 1/\alpha_i^k & & \\ & & -\alpha_{i+1}^k/\alpha_i^k & \ddots & \\ & & \vdots & & \ddots \\ & & -\alpha_p^k/\alpha_i^k & & 1 \end{pmatrix} \cdot \tag{1.21}$$

Dacă examinăm coloana i din matricea (1.21), remarcăm că ea se poate obţine din coloana k a matricei M prin următoarea regulă simplă : înlocuim în coloana k din M elementul α_i^k, din poziţie cunoscută, cu -1 şi apoi împărţim întreaga coloană la $-\alpha_i^k$. Matricea (1.21) se deosebeşte de matricea unitate prin coloana i formată după regula tocmai enunţată. Observăm că matricea (1.21) se poate memora în calculator sub forma

$$(i\,;\ -\alpha_1^k/\alpha_i^k, \dots, -\alpha_{i-1}^k/\alpha_i^k, 1/\alpha_i^k, -\alpha_{i+1}^k/\alpha_i^k, \dots, -\alpha_p^k/\alpha_i^k). \tag{1.22}$$

În continuare o matrice de forma (1.21) va fi denumită *matrice de pivotaj*.

Aşadar, un procedeu de realizare a unui pivotaj cu pivotul α_i^k asupra matricei M este următorul :

1° Se formează din coloana k a matricei M vectorul (1.22).

2° Se înmulţeşte M la stînga cu matricea (1.21).

Scanned with OKEN Scanner

Anticipăm faptul că înmulţirea matricei (1.21) cu orice coloană din M se poate face de asemenea după o regulă simplă (v. § 3.5).

Exemplul 1.3. Să reconsiderăm exemplul anterior şi să realizăm operaţiile de trecere de la matricea $\mathscr{A}$ la $\mathscr{A}^1$ şi de la matricea $\mathscr{A}^1$ la $\mathscr{A}^2$ prin procedeul tocmai expus. În coloana întîi din $\mathscr{A}$ dorim să avem coloana întîi din matricea unitate de ordinul trei. Folosind coloana întîi şi regula de formare a vectorului (1.22), obţinem vectorul (1; 1, −2, −3), care reprezintă matricea

$$P = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix}; \tag{1.23}$$

prin înmulţire directă şi comparare putem verifica egalitatea $P\mathscr{A} = \mathscr{A}^1$.

Apoi, în coloana a doua din $\mathscr{A}^1$ dorim să obţinem coloana a doua din matricea unitate de ordinul trei. Folosind coloana a doua şi regula de formare a vectorului (1.22), obţinem vectorul (2; 2/5, 1/5, −2) care reprezintă matricea

$$P' = \begin{pmatrix} 1 & 2/5 & 0 \\ 0 & 1/5 & 0 \\ 0 & -2 & 1 \end{pmatrix}; \tag{1.24}$$

prin înmulţire directă şi comparare putem verifica egalitatea $P'\mathscr{A}^1 = \mathscr{A}^2$.

Se înţelege, înainte de pivotajele succesive $\mathscr{A} \to \mathscr{A}^1$, $\mathscr{A}^1 \to \mathscr{A}^2$ a fost necesar de fiecare dată să ne convingem că în poziţiile (1, 1) şi (2, 2) respectiv aveam elemente nenule. Mai remarcăm modul de scriere a matricei finale, anume $\mathscr{A}^2 = P'(P\mathscr{A})$, sub forma $\mathscr{A}^2 = (P'P)\,\mathscr{A}$; acest mod de scriere anticipează calea cea mai simplă de obţinere directă a unei matrice rezultate dintr-o matrice dată prin mai multe pivotaje succesive (v. § 1.4.2).

1.4. INVERSAREA UNEI MATRICE

1.4.1. *EXPLICITAREA UNUI SISTEM LINIAR ÎN RAPORT CU UN GRUP DAT DE VARIABILE*

Problema explicitării unui sistem liniar dat (S^0) constă în găsirea unui sistem liniar explicitat echivalent celui dat. S-a arătat mai sus cum se poate rezolva problema prin cel mult m pivotaje succesive, cu pivoţii aleşi în cursul aplicării procedeului de explicitare. Dacă, după r pivotaje, $r \leqslant m$, era imposibil de a se găsi un nou pivot, dar sistemul liniar (S^0) era compatibil, se obţinea un sistem liniar explicitat (S^r), echivalent celui dat, cu r ecuaţii; sistemul

liniar explicitat determinat în final avea un anumit grup de r variabile principale.

Cînd am formulat problema explicitării unui sistem liniar, nu ne-a interesat care va fi grupul de variabile principale ale sistemului liniar explicitat echivalent celui dat. Mai mult, nu ne-a interesat cîte variabile fac parte din acest grup. Rezolvînd problema cu ajutorul metodei eliminării complete, nu ne-a interesat de asemenea dacă în afara sistemului liniar explicitat care era obținut nu existau și alte sisteme liniare explicitate echivalente celui dat. Acum, ne punem problema de a explicita un sistem liniar dat, în raport cu un grup prescris de m variabile principale, m fiind ca de obicei numărul ecuațiilor sistemului. Ținînd seamă de observațiile de mai sus, în cazul unui sistem liniar compatibil cu un număr de ecuații reductibile (consecințe ale celorlalte), nu vom putea face explicitarea în raport cu nici un grup de m variabile. Chiar și în cazul considerării unui sistem liniar compatibil fără ecuații reductibile s-ar putea ca explicitarea în raport cu anumite grupuri de m variabile să fie imposibilă (v. exemplul 1.4).

Deosebirea dintre problema explicitării unui sistem liniar, studiată pînă acum, și problema explicitării unui sistem liniar în raport cu un grup dat de m variabile este că în ultima problemă dorim ca, prin transformări elementare, să facem să apară toate cele m coloane ale matricei unitate de ordinul m într-un anumit grup de m coloane ale matricei lărgite a sistemului. De aceea, pentru a rezolva noua problemă, nu rămîne decît să aplicăm aceeași metodă a eliminării complete, dar cu restricția de a se căuta pivoții succesivi numai în cele m coloane fixate dinainte. Este clar că aplicarea procedeului va fi oprită atunci cînd nu vom putea găsi un nou pivot (element nenul situat într-o linie diferită de acelea ale pivoților precedenți), deși au fost cercetate toate cele m coloane. Remarcăm că o condiție necesară pentru realizarea operației de explicitare este $m \leqslant n$.

Să presupunem, ceea ce nu restrînge generalitatea, că grupul de m variabile în raport cu care se cere a se face explicitarea este $x_1, x_2, \ldots, x_m$. Să descriem pasul $r+1$ al metodei, presupunînd că în urma realizării primilor r pași, $(0 \leqslant r \leqslant m)$, s-a obținut un sistem liniar (S^r), echivalent cu sistemul liniar dat (S^0), iar matricea lărgită $\mathscr{A}^r$ a sistemului (S^r) conține în primele r coloane primele

r coloane ale matricei unitate de ordinul m, adică este de forma

$$\mathscr{A}^r = \left(\begin{array}{ccc|ccc|c} 1 & \dots & 0 & \alpha_1^{r+1} & \dots & \alpha_1^n & \beta_1 \\ \vdots & & \vdots & \vdots & & \vdots & \vdots \\ 0 & \dots & 1 & \alpha_r^{r+1} & \dots & \alpha_r^n & \beta_r \\ \vdots & & \vdots & \vdots & & \vdots & \vdots \\ 0 & \dots & 0 & \alpha_m^{r+1} & \dots & \alpha_m^n & \beta_m \end{array}\right). \tag{1.25}$$

Operaţiile pasului $r+1$ sînt următoarele:

a) Stabilim dacă $r = m$ sau $r < m$; în primul caz procedeul este oprit, sistemul liniar (S^r) este explicitat în raport cu primele m variabile; în caz contrar, trecem la operaţia următoare.

b) Cercetăm elementele coloanei $r+1$ situate în liniile $r+1, \dots, m$, pînă descoperim un element nenul; dacă toate aceste elemente sînt nule, procedeul este oprit, sistemul liniar dat (S^0) nu poate fi explicitat în raport cu grupul de m variabile prestabilite; dacă am găsit un element nenul, atunci ori el este deja în poziţia $(r+1, r+1)$ ori îi vom aduce în această poziţie printr-o transformare (T_2) şi-l vom considera drept pivot.

c) Realizăm pivotajul cu elementul ales ca pivot şi trecem la pasul următor.

Să justificăm afirmaţia de la operaţia b). Prin orice transformare elementară un determinant oarecare de ordinul m din matricea $\mathscr{A}^r$ se va înmulţi cu un factor diferit de zero; deci, determinantul format din primele m coloane ale matricei sistemului dat va rămîne nul sau nenul, aşa cum a fost iniţial. În cazul cînd prin operaţia b) se stabileşte $\alpha_{r+1}^{r+1} = \dots = \alpha_m^{r+1} = 0$, dezvoltîndu-se determinantul primelor m coloane din $\mathscr{A}^r$ după coloana r, se constată că acesta este nul; prin urmare, determinantul cu aceeaşi poziţie din matricea iniţială a fost de asemenea nul. Dacă ar fi fost posibil a se explicita sistemul liniar în raport cu primele m variabile, am fi obţinut în final un determinant egal cu unu, deci nenul, aşa încît explicitarea sistemului în raport cu acest grup de variabile este imposibilă.

Exemplul 1.4. Să explicităm sistemul liniar

$$\left.\begin{aligned} x + y + z + u &= 1 \\ 2x - y \quad\quad - 2u &= 2 \\ \alpha x - 2y + \beta z - 3u &= 1 \end{aligned}\right\} \tag{1.26}$$

în raport cu grupul de variabile (x, y, u); α și β sînt parametri reali. Vom realiza trei pivotaje, elementele pivot fiind cele încercuite.

Pasul 1. Facem pivotajul

$$\left(\begin{array}{cccc|c} \textcircled{1} & 1 & 1 & 1 & 1 \\ 2 & -1 & 0 & -2 & 2 \\ \alpha & -2 & \beta & -3 & 1 \end{array}\right) \Rightarrow \left(\begin{array}{cccc|c} 1 & 1 & 1 & 1 & 1 \\ 0 & -3 & -2 & -4 & 0 \\ 0 & -\alpha-2 & -\alpha+\beta & -\alpha-3 & -\alpha+1 \end{array}\right).$$

Pasul 2. Facem un al doilea pivotaj

$$\left(\begin{array}{cccc|c} 1 & 1 & 1 & 1 & 1 \\ 0 & \textcircled{-3} & -2 & -4 & 0 \\ 0 & -\alpha-2 & -\alpha+\beta & -\alpha-3 & -\alpha+1 \end{array}\right) \Rightarrow \left(\begin{array}{cccc|c} 1 & 0 & 1/3 & -1/3 & 1 \\ 0 & 1 & 2/3 & 4/3 & 0 \\ 0 & 0 & -\frac{1}{3}\alpha+\beta+\frac{4}{3} & \frac{1}{3}\alpha-\frac{1}{3} & -\alpha+1 \end{array}\right)$$

Pasul 3. Acum, dacă $\alpha \neq 1$, putem lua ca pivot $\frac{1}{3}(\alpha-1)$ și obținem sistemul liniar explicitat în raport cu (x, y, u) printr-un al treilea pivotaj, anume

$$\left(\begin{array}{cccc|c} 1 & 0 & 1/3 & -1/3 & 1 \\ 0 & 1 & 2/3 & 4/3 & 0 \\ 0 & 0 & -\frac{1}{3}\alpha+\beta+\frac{4}{3} & \textcircled{\frac{1}{3}\alpha-\frac{1}{3}} & -\alpha+1 \end{array}\right) \Rightarrow \left(\begin{array}{cccc|c} 1 & 0 & \frac{\beta+1}{\alpha-1} & 0 & 0 \\ 0 & 1 & \frac{2\alpha-4\beta-6}{\alpha-1} & 0 & 4 \\ 0 & 0 & \frac{-\alpha+3\beta+4}{\alpha-1} & 1 & -3 \end{array}\right)$$

Dacă $\alpha = 1$, matricea obținută după al doilea pivotaj este

$$\left(\begin{array}{cccc|c} 1 & 0 & 1/3 & -1/3 & 1 \\ 0 & 1 & 2/3 & 4/3 & 0 \\ 0 & 0 & \beta+1 & 0 & 0 \end{array}\right),$$

încît sistemul (1.26) nu poate fi explicitat în raport cu grupul de variabile (x, y, u). Totuși, dacă $\beta \neq -1$, sistemul poate fi explicitat în raport cu grupul de variabile (x, y, z) prin pivotajul

$$\left(\begin{array}{cccc|c} 1 & 0 & 1/3 & -1/3 & 1 \\ 0 & 1 & 2/3 & 4/3 & 0 \\ 0 & 0 & \textcircled{\beta+1} & 0 & 0 \end{array}\right) \Rightarrow \left(\begin{array}{cccc|c} 1 & 0 & 0 & -1/3 & 1 \\ 0 & 1 & 0 & 4/3 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{array}\right),$$

în timp ce, dacă $\beta = -1$, explicitarea sistemului (1.26) este imposibilă în raport cu orice grup de trei variabile.

Scanned with OKEN Scanner

1.4.2. *METODE DE INVERSARE A UNEI MATRICE*

Fie A o matrice pătrată de ordinul m și $\mathscr{A} = (A|E)$ matricea obținută prin alăturarea matricei unitate de ordinul m la matricea A. Se știe că se numește matrice inversă a unei matrice pătrate A o altă matrice, notată cu A^{-1}, astfel încît

$$AA^{-1} = A^{-1}A = E. \tag{1.27}$$

Evident, A^{-1} trebuie să fie tot o matrice pătrată, de același ordin cu A.

Să presupunem că nu se știe dacă pentru matricea A dată există sau nu matrice inversă și ne punem problema de a stabili fie inexistența inversei, fie matricea inversă în cazul cînd aceasta există. Deși, cel puțin la prima vedere, nu pare a fi vreo legătură între problema tocmai formulată și problema explicitării unui sistem liniar în raport cu un grup dat de variabile, și nici nu vom face apel la vreo asemenea legătură, vom putea totuși rezolva noua problemă folosind aceeași metodă.

Să presupunem că aplicăm matricei $\mathscr{A}$ un procedeu de explicitare în raport cu primele m variabile, ca și cum $\mathscr{A}$ ar fi matricea lărgită a unui sistem liniar. Desigur, putem să folosim fie pivotaje realizate prin transformări elementare, fie pivotaje realizate prin înmulțirea la stînga cu matrice elementare. Scopul final este deci de a se obține în matricea finală matricea unitate de ordinul m dispusă în poziția unde se afla inițial A. Aplicarea procedeului poate duce la una dintre următoarele două situații: 1° procedeul este oprit în cursul operației b) a unui pas $r+1$ și nu vom putea realiza în coloana $r+1$, coloana $r+1$ din matricea unitate de ordinul m; 2° am obținut în locul unde inițial se afla matricea A matricea unitate de ordinul m.

În primul caz, așa cum s-a arătat în § 1.4.1 pe parcursul justificării procedeului, avem $\det A = 0$, or se știe că o matrice pătrată cu determinant nul nu are inversă; prin urmare, dacă aplicarea procedeului duce la situația 1°, problema considerată este rezolvată în sensul stabilirii inexistenței inversei.

În cazul al doilea, ținînd seamă de faptul că orice succesiune de pivotaje se poate realiza și prin înmulțirea succesivă la stînga cu matrice de pivotaj, adică de forma (1.21), și, notînd cu M produsul acestor matrice de pivotaj luate în ordine inversă, obținem

$$M\mathscr{A} = (MA|ME) = (E|M). \tag{1.28}$$

Scanned with OKEN Scanner

Deoarece $MA = E$, matricea M este inversă lui A și, prin urmare, dacă aplicarea procedeului duce la situația 2°, problema considerată este rezolvată în sensul găsirii matricei inverse lui A, anume inversa se găsește la sfîrșitul calculelor în poziția ocupată inițial în $\mathscr{A}$ de matricea E.

Așadar, pentru rezolvarea problemei aflării inversei unei matrice date A, vom folosi metoda de calcul descrisă în § 1.4.1 pentru explicitarea unui sistem liniar, aplicată matricei $\mathscr{A} = (A|E)$, pivotajele fiind realizate fie direct prin transformări elementare fie prin înmulțiri la stînga succesive cu matrice elementare. Ne este totuși la îndemînă și o altă metodă, denumită *metoda formei de produs a inversei.*

Să notăm cu $M_1, M_2, \ldots, M_m$ matricele de pivotaj care construiesc succesiv în primele m coloane din A coloanele matricei unitate de ordinul m. Cum $M = A^{-1}$ și $M = M_m \ldots M_1$, rezultă că inversa matricei A se scrie ca produs al matricelor de pivotaj, anume

$$A^{-1} = M_m \ldots M_1. \tag{1.29}$$

Folosind procedeul de determinare a matricelor de pivotaj dat în § 1.3, putem construi M_1 din prima coloană a lui A, apoi M_2 din a doua coloană a lui M_1A, ..., în sfîrșit M_m din a m-a coloană a lui $M_{m-1} \ldots M_1A$. Se înțelege, după construirea lui M_1 nu mai este necesară prima coloană din A, după construirea lui M_2 nu mai este necesară a doua coloană din A, ..., încît putem construi inversa matricei A pe baza formulei (1.29) cu mare economie de memorie a calculatorului.

Mai remarcăm că inversarea unei matrice este o problemă conexă problemei stabilirii balanței legăturilor între ramurile unei economii [3] și are, în consecință, o mare valoare practică.

Exemplul 1.5. Să determinăm inversa matricei

$$A = \begin{pmatrix} 1 & 2 \\ 1 & 1 \end{pmatrix}. \tag{1.30}$$

Considerăm matricea

$$\mathscr{A} = \left(\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ -1 & 1 & 0 & 1 \end{array}\right). \tag{1.31}$$

și realizăm prin transformări elementare pivotajele

$$\left(\begin{array}{cc|cc} \textcircled{1} & 2 & 1 & 0 \\ -1 & 1 & 0 & 1 \end{array}\right) \Rightarrow \left(\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 0 & 3 & 1 & 1 \end{array}\right),$$

$$\left(\begin{array}{cc|cc} 1 & 2 & 1 & 0 \\ 0 & \textcircled{3} & 1 & 1 \end{array}\right) \Rightarrow \left(\begin{array}{cc|cc} 1 & 0 & 1/3 & -2/3 \\ 0 & 1 & 1/3 & 1/3 \end{array}\right);$$

sîntem în situația 2°, deci inversa lui A este matricea

$$A^{-1} = \begin{pmatrix} 1/3 & -2/3 \\ 1/3 & 1/3 \end{pmatrix}. \tag{1.32}$$

Cele două pivotaje se realizează și prin înmulțirea lui A la stînga succesiv cu matricele de pivotaj

$$M_1 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \qquad M_2 = \begin{pmatrix} 1 & -2/3 \\ 0 & 1/3 \end{pmatrix}, \tag{1.33}$$

dintre care M_1 este construită din prima coloană a lui A, iar M_2 este construită din a doua coloană a lui M_1A. Analizînd însă elementele problemei care trebuie păstrate în memoria calculatorului în acest caz, ajungem la concluzia că metoda formei de produs a inversei este mult mai economică.

Din prima coloană a matricei A formăm matricea M_1, reprezentată în memorie prin vectorul (1 ; 1, 1) de forma (1.22), și simultan ștergem această primă coloană. Realizăm produsul lui M_1 cu a doua coloană din A, formăm din rezultat matricea M_2, reprezentată în memorie prin vectorul (2 ; −2/3, 1/3), și simultan ștergem această a doua coloană. Cum $A^{-1}=M_2M_1$, efectuăm acest produs și obținem matricea inversă. Se înțelege că la fiecare pas trebuie căutat pivotul și adus în poziția corespunzătoare, dacă el nu se găsește acolo. Remarcăm că o regulă simplă realizează produsul unei matrice de pivotaj cu o altă matrice, regulă care reclamă doar cunoașterea vectorului reprezentativ al matricei de pivotaj (v. § 3.5). Această regulă va fi formulată și folosită la elaborarea formelor revizuite ale algoritmilor simpleciali.

1.5. SOLUȚII DE BAZĂ ALE UNUI SISTEM LINIAR

Fie (S) un sistem liniar de m ecuații cu n necunoscute, despre care presupunem că poate fi explicitat în raport cu cel puțin un grup de m variabile. Vom numi *formă explicită* a sistemului (S) orice sistem liniar, echivalent cu (S), a cărui matrice conține coloanele matricei unitate de ordinul m. În § 1.2 și § 1.3 am arătat cum poate fi găsită prin metoda eliminării complete o formă explicită a sistemului (S); în § 1.4 am arătat cum se poate obține o formă explicită a sistemului (S) cu un grup de variabile principale prescrise, dacă aceasta există. Deoarece cu cele n variabile ale sistemului (S) putem forma C_n^m grupuri diferite de cîte m variabile,

este clar că un sistem liniar, care poate fi explicitat în raport cu cel puţin un grup de m variabile, va avea cel mult C_n^m forme explicite. Acest număr maxim va fi atins dacă explicitarea poate avea loc în raport cu oricare dintre cele C_n^m grupuri diferite de cîte m variabile şi toate formele explicite sînt distincte. Prin urmare, am demonstrat teorema următoare :

Teorema 1.4. *Un sistem liniar, care admite cel puţin o formă explicită cu m variabile principale, va admite cel mult C_n^m forme explicite, unde n este numărul variabilelor şi m numărul ecuaţiilor sistemului.*

Dacă (S') este o formă explicită a sistemului liniar (S) cu variabilele principale avînd indicii într-o mulţime I' şi $I'' = I' - \{i\} + \{k\}$, $i \in I'$, $k \notin I'$, vom putea stabili simplu fie inexistenţa unei forme explicite a lui (S) cu variabilele principale avînd indicii în I'', fie însăşi această formă explicită în caz contrar. Anume, dacă elementul din poziţia (i, k) din matricea lui (S') este nul, am probat inexistenţa formei explicite căutate, iar dacă acest element este nenul, vom găsi forma explicită printr-un pivotaj cu elementul considerat drept pivot. Din aceste observaţii rezultă că vom putea trece prin pivotaj de la o formă explicită la orice altă formă explicită diferind de prima printr-o variabilă principală, dacă acea formă explicită există, sau să stabilim inexistenţa ei în caz contrar. Pe această cale vom putea construi prin pivotaje succesive lista tuturor formelor explicite ale sistemului liniar dat (S).

Exemplul 1.6. Fie sistemul liniar

$$\left.\begin{aligned} x \quad + z - u &= \lambda \\ y - z + u &= 2 \end{aligned}\right\} \tag{S}$$

cu matricea lărgită

$$\mathscr{A} = \left(\begin{array}{cccc|c} 1 & 0 & 1 & -1 & \lambda \\ 0 & 1 & -1 & 1 & 2 \end{array}\right). \tag{1.34}$$

Sistemul se prezintă chiar sub formă explicită, în raport cu grupul de variabile (x, y). Putem obţine prin pivotaje din (1.34) formele explicite în raport cu grupurile de variabile (z, y), (u, y) şi (x, z), (x, u). Pivotajele

$$\left(\begin{array}{cccc|c} 1 & 0 & \textcircled{1} & -1 & \lambda \\ 0 & 1 & -1 & 1 & 2 \end{array}\right) \Rightarrow \left(\begin{array}{cccc|c} 1 & 0 & 1 & -1 & \lambda \\ 1 & 1 & 0 & 0 & \lambda + 2 \end{array}\right)$$

$$\left(\begin{array}{cccc|c} 1 & 0 & 1 & \textcircled{-1} & \lambda \\ 0 & 1 & -1 & 1 & 2 \end{array}\right) \Rightarrow \left(\begin{array}{cccc|c} -1 & 0 & -1 & 1 & -\lambda \\ 1 & 1 & 0 & 0 & \lambda + 2 \end{array}\right)$$

și

$$\left(\begin{array}{cccc|c} 1 & 0 & 1 & -1 & \lambda \\ 0 & 1 & \textcircled{-1} & 1 & 2 \end{array}\right) \Rightarrow \left(\begin{array}{cccc|c} 1 & 1 & 0 & 0 & \lambda+2 \\ 0 & -1 & 1 & -1 & -2 \end{array}\right),$$

$$\left(\begin{array}{cccc|c} 1 & 0 & 1 & -1 & \lambda \\ 0 & 1 & -1 & \textcircled{1} & 2 \end{array}\right) \Rightarrow \left(\begin{array}{cccc|c} 1 & 1 & 0 & 0 & \lambda+2 \\ 0 & 1 & -1 & 1 & 2 \end{array}\right),$$

ne conduc la formele explicite

$$\left.\begin{array}{l} x \quad +z-u=\lambda \\ x+y \quad =\lambda+2 \end{array}\right\}, \qquad \left.\begin{array}{l} -x \quad -z+u=-\lambda \\ x+y \quad =\lambda+2 \end{array}\right\}, \tag{1.35}$$

și

$$\left.\begin{array}{l} x+y \quad =\lambda+2 \\ -y+z-u=-2 \end{array}\right\}, \qquad \left.\begin{array}{l} x+y \quad =\lambda+2 \\ y-z+u=2 \end{array}\right\}. \tag{1.36}$$

Se înțelege, am putea încerca să obținem și a șasea formă explicită a sistemului (S), în raport cu grupul de variabile (z, u), plecînd de la oricare dintre matricele lărgite ale formelor explicite (1.35), (1.36), dar aceasta nu există. Cum $m = 2$, $n = 4$ și $C_4^2 = 6$, am obținut toate formele explicite ale sistemului liniar dat.

Se numește *soluție de bază* a unui sistem liniar (S) orice soluție a sistemului obținută prin egalarea cu zero a variabilelor secundare și determinarea valorilor variabilelor principale, într-o formă explicită a lui (S).

În baza definiției, oricărei forme explicite îi corespunde în mod unic o soluție de bază a lui (S); desigur, este posibil ca la două forme explicite diferite să corespundă aceeași soluție de bază a lui (S). Din teorema 1.4 rezultă atunci teorema următoare:

Teorema 1.5. *Un sistem liniar, care are cel puțin o soluție de bază, va avea cel mult C_n^m soluții de bază, unde n este numărul variabilelor și m este numărul ecuațiilor sistemului.*

Exemplul 1.7. Sistemul liniar (S) considerat în exemplul precedent are soluțiile de bază: $x = \lambda$, $y = 2$, $z = 0$, $u = 0$ obținută din forma explicită inițială; $x = 0$, $y = \lambda + 2$, $z = \lambda$, $u = 0$ și $x = 0$, $y = \lambda + 2$, $z = 0$, $u = -\lambda$ obținute din formele explicite (1.35); $x = \lambda + 2$, $y = 0$, $z = -2$, $u = 0$ și $x = \lambda + 2$, $y = 0$, $z = 0$, $u = 2$ obținute din formele explicite (1.36)

Observăm că dacă în partea stîngă a fiecărei linii din matricea lărgită a unei forme explicite scriem variabila principală din ecuația corespunzătoare acelei linii, vom putea citi direct în ultima coloană a matricei valorile variabilelor principale din soluția de bază corespunzătoare formei explicite. Pivotajele care realizează

trecerea de la o formă explicită la alta realizează în același timp și trecerea de la o soluție de bază la alta. Desigur, odată cu pivotajul trebuie să notăm în stînga liniei corespunzătoare noua variabilă principală.

Din definiția dată pentru conceptul de soluție de bază a unui sistem liniar rezultă că, într-o soluție de bază a unui sistem de m ecuații cu n necunoscute, cel puțin $n - m$ variabile sînt nule sigur și anume variabilele secundare în număr de $n - m$. Dacă într-o soluție de bază sînt exact $n - m$ variabile nule, vom spune că soluția de bază este *nedegenerată*. Dacă mai mult de $n - m$ variabile sînt nule într-o soluție de bază, vom spune că soluția de bază este *degenerată*.

Soluțiile de bază ale unui sistem liniar se clasifică și după un alt criteriu. Dacă orice variabilă are într-o soluție de bază o valoare nenegativă, vom spune că soluția de bază respectivă este *admisibilă*. În caz contrar, vom spune că soluția de bază nu este admisibilă.

Exemplul 1.8. Sistemul liniar (S) considerat în exemplele precedente are, pentru diversele valori ale parametrului real λ, soluții de bază nedegenerate și degenerate, admisibile și neadmisibile, cum rezultă din tabelul următor :

λ	S_1	S_2	S_3	S_4	S_5
$\lambda<-2$	ned., nead.	ned., nead.	ned., nead.	ned., nead.	ned., nead.
-2	,,	d., nead.	d., ad.	d., nead.	d., ad.
$-2<\lambda<0$	,,	ned., nead.	ned., ad.	ned., nead.	ned., ad.
0	d., ad.	d., ad.	d., ad.	,,	,,
$\lambda>0$	ned., ad.	ned., ad.	ned., nead.	,,	,,

unde s-au notat soluțiile de bază cu

$$S_1 : x = \lambda, \quad y = 2, \quad z = 0, \quad u = 0;$$
$$S_2 : x = 0, \quad y = \lambda + 2, \quad z = \lambda, \quad u = 0;$$
$$S_3 : x = 0, \quad y = \lambda + 2, \quad z = 0, \quad u = -\lambda;$$
$$S_4 : x = \lambda + 2, \quad y = 0, \quad z = -2, \quad u = 0;$$
$$S_5 : x = \lambda + 2, \quad y = 0, \quad z = 0, \quad u = 2.$$

Pentru $\lambda = -2$ avem numai trei soluții de bază distincte, căci S_2 cu S_4 și S_3 cu S_5 se confundă, iar pentru $\lambda = 0$ avem tot trei soluții de bază distincte, căci S_1, S_2, S_3 sînt confundate.

Scanned with OKEN Scanner

2

PROBLEMA GENERALĂ A PROGRAMĂRII LINIARE: FORMULAREA PROBLEMEI, EXPLICITAREA UNUI PROGRAM LINIAR, PIVOTAJ PRIMAL ŞI DUAL

Programarea liniară este una dintre disciplinele matematice cu aplicaţii practice imediate; alături de alte discipline ca teoria grafurilor, teoria jocurilor, teoria aşteptării etc., programarea liniară face parte dintr-un domeniu ştiinţific destul de greu de definit, cunoscut sub denumirea de cercetare operaţională.

Prima metodă generală de rezolvare a programelor liniare este metoda simplex primală, dată de G. B. Dantzig [4]; o a doua metodă generală, metoda simplex duală, este datorată lui C. E. Lemke [5]. După elaborarea acestor două metode atenţia cercetărilor s-a îndreptat către clase de probleme la a căror rezolvare se pot folosi cele două metode şi teoria s-a îmbogăţit cu capitole noi, dintre care remarcabile din punctul de vedere al rezultatelor au devenit capitolele cu privire la problemele de programare în numere întregi şi problemele de transport. Mai tîrziu a crescut atît diversitatea problemelor, cît şi a metodelor de rezolvare, precum şi numărul aplicaţiilor publicate; de curînd au fost publicate şi programe pentru calculator, pentru diferiţi algoritmi aparţinînd programării liniare.

Deoarece programarea liniară s-a dezvoltat în contact permanent cu practica, unele concepte au denumiri ce relevă tocmai această legătură; de aceea, ni se pare important a începe dezvoltarea teoriei cu trei dintre modelele matematice ale unor probleme practice, modele care sînt programe liniare şi a căror imagine va justifica în continuare denumirile introduse.

2.1. TREI MODELE

2.1.1. *PROBLEMA FOLOSIRII OPTIME A RESURSELOR*

Să considerăm un proces de producţie în cursul căruia se realizează n articole P_k, $(k = 1, \ldots, n)$; procesul se desfăşoară pe baza unui disponibil de m resurse R_i, $(i = 1, \ldots, m)$, limitate superior

de cantităţile b_i, $(i = 1, \ldots, m)$. Se cunoaşte matricea $A = (a_i^k)$, în care sînt date cantităţile a_i^k, cu semnificaţia: a_i^k = cantitatea din resursa R_i necesară pentru producerea unei unităţi din articolul P_k, $(i = 1, \ldots, m;\ k = 1, \ldots, n)$. Se mai cunosc beneficiile unitare c_k pentru articolele P_k, $(k = 1, \ldots, n)$. Se cere a se determina cantităţile x_k de produs P_k, $(k = 1, \ldots, n)$, care urmează a se realiza în procesul considerat, cu resurse limitate, astfel ca procesul să aibă drept rezultat beneficiul maxim. Problema tocmai formulată economic mai este denumită şi problema sortimentului optim. Se poate trece imediat de la aspectul economic la cel matematic al problemei, scriind sub formă algebrică restricţiile economice. Aceste restricţii sînt de trei feluri: restricţii datorate limitării resurselor, restricţii impuse de sensul economic al variabilelor şi restricţia de maximizare a beneficiului.

Restricţiile de limitare a resurselor exprimă faptul că disponibilul pentru fiecare resursă R_i nu poate fi întrecut de cantitatea consumată în procesul de producţie. Cum x_k reprezintă mărimi economice ale căror valori negative ar fi lipsite de sens, variabilele problemei trebuie să fie nenegative. Traducerea algebrică a restricţiilor din primele două categorii este

$$\sum_{k=1}^{n} a_i^k x_k \leqslant b_i, \quad (i = 1, \ldots, m),$$
$$x_k \geqslant 0, \quad (k = 1, \ldots, n). \tag{2.1}$$

Beneficiul total, obţinut ca sumă a beneficiilor pentru fiecare articol în parte, este o funcţie de cantităţile produse, anume

$$f = \sum_{k=1}^{n} c_k x_k \tag{2.2}$$

care se cere a fi maximizată.

Problema considerată se formulează acum matematic astfel: în mulţimea soluţiilor sistemului de inecuaţii (2.1) să se găsească o soluţie ce dă valoarea maximă a funcţiei f. Se înţelege, problema are sens numai dacă sistemul (2.1) are cel puţin o soluţie, încît

Scanned with OKEN Scanner

metoda generală de rezolvare, aplicată unui sistem a cărui compatibilitate n-a fost studiată în prealabil, va trebui fie să stabilească incompatibilitatea sistemului, fie, în caz de compatibilitate, să determine soluția căutată.

2.1.2. *PROBLEMA AMESTECULUI OPTIM*

Să presupunem că pentru realizarea unui amestec se pot întrebuința n materii prime M_k, $(k = 1, \ldots, n)$, disponibile în cantități nelimitate. Amestecul trebuie să conțină m substanțe S_i, $(i = 1, \ldots, m)$, care se găsesc în materiile prime, în așa fel ca fiecare substanță S_i să intre în unitatea de amestec în cantitatea cunoscută b_i, $(i = 1, \ldots, m)$. Se cunoaște matricea $A = (a_i^k)$, în care sînt date cantitățile a_i^k cu semnificația: a_i^k = cantitatea de substanță S_i ce trece în amestec dintr-o unitate de materie primă M_k, $(i = 1, \ldots, m;\ k = 1, \ldots, n)$. Se mai cunosc costurile unitare c_k pentru materiile prime M_k, $(k = 1, \ldots, n)$. Se cere a se determina cantitățile x_k de materie primă M_k, $(k = 1, \ldots, n)$, necesare pentru a se obține unitatea de amestec cu compoziție prescrisă, la costul total minim. Problema tocmai formulată economic mai este denumită și problema dietei, datorită modului de interpretare ce poate fi dat termenilor de „materie primă" și „substanță" într-o problemă de formare a unei rații alimentare cu cost minim.

Restricțiile sînt și acum de trei feluri: restricțiile de compoziție a amestecului, restricțiile datorate sensului economic al variabilelor și restricția de minimizare a costului.

Restricțiile tehnologice de compoziție a amestecului exprimă faptul că, pentru fiecare substanță S_i, cantitatea de materie primă folosită produce în unitatea de amestec exact cantitatea de substanță prescrisă. Restricțiile datorate sensului economic al variabilelor sînt analoage celor din modelul precedent. Traducerea algebrică a restricțiilor din primele două categorii este

$$\sum_{k=1}^{n} a_i^k x_k = b_i \quad (i = 1, \ldots, m),$$
$$x_k \geqslant 0, \quad (k = 1, \ldots, n). \tag{2.3}$$

Costul total al materiei prime necesare pentru producerea unității de amestec este dat de funcția

$$f = \sum_{k=1}^{n} c_k x_k, \tag{2.4}$$

care se cere a fi minimizată.

Problema considerată se formulează matematic astfel: în mulțimea soluțiilor sistemului (2.3) să se găsească o soluție ce dă valoarea minimă a funcției f. Și în acest model matematic problema are sens numai dacă sistemul (2.3) este compatibil; de aceea, o metodă de rezolvare va trebui să obțină fie rezolvarea, în caz de compatibilitate, fie stabilirea incompatibilității.

2.1.3. *PROBLEMA TRANSPORTURILOR*

Să presupunem că un anumit produs este stocat în m centre de depozitare D_i în cantitățile a_i, $(i = 1, \ldots, m)$, și în n centre de consum C_k este cerut acest produs în cantitățile b_k, $(k = 1, \ldots, n)$. Se cunoaște matricea $C = (c_{ik})$ în care sînt date numerele $c_{ik} \geqslant 0$ cu semnificația: c_{ik} = costul de transport al unității de produs de la depozitul D_i la centrul de consum C_k, $(i = 1, \ldots, m;$ $k = 1, \ldots, n)$. Se cere a se determina cantitățile x_{ik} de produs care urmează a fi repartizate de la centrele de depozitare la centrele de consum astfel ca disponibilul să fie epuizat, în fiecare depozit, cererea să fie satisfăcută exact, în fiecare centru de consum, iar costul total al transportului produsului să fie minim. Se presupune că $\sum_{i=1}^{m} a_i = \sum_{k=1}^{n} b_k$, adică disponibilul total din depozite este egal cu cererea totală a centrelor de consum. Problema tocmai formulată economic este denumită problema de transport echilibrată, datorită obiectivului urmărit și faptului că se presupune echilibrul cererii cu disponibilul. Desigur, cazul echilibrat este un caz limită, dar acesta este important, căci în teoria problemei de transport se arată că o problemă de transport neechilibrată se reduce la rezolvarea unei probleme de transport echilibrate.

Restricțiile problemei sînt din nou de trei tipuri: restricțiile asupra disponibilului și cererii, restricțiile datorate sensului economic al variabilelor și restricția de minimizare a costului.

Scanned with OKEN Scanner

Restricţiile asupra disponibilului şi cererii exprimă faptul că repartiţia căutată epuizează disponibilul fiecărui depozit D_i şi satisface exact cererea fiecărui centru de consum, C_k. Restricţiile datorate sensului economic al variabilelor sînt analoage celor din modelele precedente. Traducerea algebrică a restricţiilor din primele două categorii este

$$\begin{aligned} &\sum_{k=1}^{n} x_{ik} = a_i, \quad (i = 1, \ldots, m), \\ &\sum_{i=1}^{m} x_{ik} = b_k, \quad (k = 1, \ldots, n), \\ &x_{ik} \geqslant 0, \quad (i = 1, \ldots, m;\ k = 1, \ldots, n). \end{aligned} \tag{2.5}$$

Costul total al transportului este dat de funcţia

$$f = \sum_{i=1}^{m} \sum_{k=1}^{n} c_{ik}\, x_{ik}, \tag{2.6}$$

care se cere a fi minimizată.

Observăm că, în timp ce primele două modele au un caracter suficient de general din punct de vedere matematic, ultimul model are un aspect particular. Totuşi, datorită multiplelor proprietăţi ale sistemului (2.5) problema de transport reprezintă un important domeniu de studiu în programarea liniară. Dealtfel, această problemă a generat un număr apreciabil de probleme mai generale, fapt care a determinat închegarea unui domeniu separat intitulat teoria reţelelor de transport, domeniu considerat în unele lucrări drept un capitol important al teoriei grafurilor.

Se impune o scurtă discuţie asupra celor trei modele prezentate. În cele precedente, am înţeles prin model matematic nimic altceva decît formularea matematică a problemei economice considerate. De obicei termenul de model are un alt sens, anume acela al reprezentării sub o formă simplificată a unei clase întregi de probleme, modelul conţinînd doar elementele comune ale tuturor problemelor din acea clasă. Am putea spune că cele trei probleme considerate mai sus, fie sub formă economică, fie sub formă matematică, sînt numai modele şi în acest sens, deoarece fiecare problemă în parte nu reprezintă o anumită problemă practică numerică, dar este foarte

probabil ca diversele probleme practice de același tip să conțină printre condițiile lor restrictive cele cuprinse în formulările respective. Ceea ce mai vrem să subliniem este faptul că, deși este greu ca o problemă practică dată să se potrivească exact cu un model, totuși este util a se cunoaște un număr de modele, cît și metodele lor specifice de rezolvare, pentru ca prin aceasta fiind în fața unei anumite probleme practice să avem o experiență ce ne poate ajuta la formularea și rezolvarea problemei noastre.

2.2. FORMULAREA PROBLEMEI GENERALE A PROGRAMĂRII LINIARE

Să considerăm un sistem liniar (S) care poate conține fie numai ecuații, fie numai inecuații, fie și ecuații și inecuații. Un asemenea sistem îl putem scrie sub forma

$$\sum_{k=1}^{n} a_i^k x_k \top b_i, \quad (i = 1, \ldots, m), \tag{2.7}$$

unde semnul $\top$ reprezintă semnul $=$, ori $\leqslant$, ori $\geqslant$, în diferitele condiții (2.7). Considerăm și sistemul de condiții

$$x_k \geqslant 0, \quad (k = 1, \ldots, n). \tag{2.8}$$

Introducem o funcție liniară

$$f = \sum_{k=1}^{n} c_k x_k. \tag{2.9}$$

Problema generală a programării liniare se formulează astfel: în mulțimea soluțiilor sistemului (2.7), (2.8) să se găsească o soluție ce dă valoarea maximă (sau minimă) a funcției (2.9).

Este clar că sistemul liniar (2.7) poate fi scris sub forma matriceală

$$AX \top b, \tag{2.10}$$

condițiile (2.8) sub forma

$$X \geqslant 0 \tag{2.11}$$

și funcția (2.9) se transcrie

$$f = c'X, \tag{2.12}$$

Scanned with OKEN Scanner

unde $A = (a_i^k)$, $b = (b_i)$, $c = (c_k)$, $X = (x_k)$, reprezintă matricea $m \times n$ a coeficienţilor variabilelor, vectorul m-dimensional al termenilor liberi, vectorul n-dimensional al coeficienţilor funcţiei liniare şi vectorul n-dimensional al variabilelor, respectiv.

Sistemul (2.7), sau (2.10), este denumit *sistemul restricţiilor* problemei, iar orice soluţie a sa, soluţie a problemei. Sistemul (2.8), sau (2.11), este denumit *sistemul condiţiilor de admisibilitate* sau nenegativitate, iar orice soluţie a sistemelor (2.7), (2.8), *soluţie admisibilă* a problemei. Funcţia (2.9), sau (2.12), este denumită *funcţia obiectiv*, sau funcţia economică, sau criteriul de optimizare. O soluţie admisibilă a problemei care dă valoarea maximă (sau minimă) a funcţiei obiectiv pe mulţimea soluţiilor admisibile este denumită *soluţie optimă* a problemei. Aşadar, problema generală a programării liniare este : să se găsească o soluţie optimă a sistemului de condiţii :

$$AX \top b, \quad X \geqslant 0, \quad f = c'X = \max \text{ (min)}. \tag{2.13}$$

Remarcăm că ar fi posibil ca valoarea maximă (sau minimă) a funcţiei f, pe mulţimea soluţiilor admisibile ale problemei, să fie atinsă pentru mai multe soluţii ; conform formulării precedente, se caută numai una dintre aceste soluţii optime. Evident, am putea defini ca problemă generală a programării liniare aceea de a se determina întreaga mulţime a soluţiilor optime. Dealtfel, lucrările consacrate teoriei programării liniare optează aproximativ în aceeaşi măsură pentru una sau cealaltă dintre cele două formulări. În cele precedente s-a optat pentru prima formulare, deoarece în întreaga lucrare nu vom da decît metode de aflare a unei singure soluţii optime. Se înţelege că practic este deosebit de util de a se cunoaşte toate soluţiile optime, argument pe care-l aduc cei ce optează pentru a doua formulare.

Problema generală a programării liniare este, din punct de vedere matematic, o problemă de extrem legat: totuşi, faţă de problemele de extrem legat considerate în analiza matematică sînt două deosebiri esenţiale. Mai întîi, că printre legăturile la care sînt supuse variabilele funcţiei de extremizat există şi inecuaţii ; chiar dacă sistemul restricţiilor problemei conţine numai ecuaţii, mai rămîn condiţiile de admisibilitate. Apoi, într-o problemă de programare liniară nu se cere un extrem relativ, ci un extrem absolut. Aceste două deosebiri au impus căutarea unor noi metode de re-

Scanned with OKEN Scanner

zolvare pentru problemele de programare liniară, diferite de cele din analiza clasică.

Sistemul de condiţii (2.13) arată că există două tipuri de probleme de programare liniară : probleme de maximizare şi probleme de minimizare. Totuşi, în continuare vom dezvolta numai teoria problemelor de minimizare, căci pentru a se afla o soluţie optimă a unei probleme de maximizare, relativă la o funcţie obiectiv f, este suficient a se afla o soluţie optimă a unei probleme de minimizare relativă la funcţia obiectiv $f' = -f$. Se înţelege că dacă în acelaşi timp căutăm şi valoarea maximă a lui f, aceasta va fi obţinută schimbînd semnul valorii minime a lui f'.

O altă observaţie permite încă a se simplifica teoria, considerîndu-se, fără a se restrînge generalitatea, sistemul de restricţii al problemei generale format numai din ecuaţii. Într-adevăr, dacă în sistemul restricţiilor unui program liniar avem o inecuaţie de forma

$$\sum_{k=1}^{n} \alpha_k x_k \leqslant \beta, \tag{2.14}$$

putem înlocui această inecuaţie prin condiţiile echivalente

$$\sum_{k=1}^{n} \alpha_k x_k + y = \beta, \quad y \geqslant 0. \tag{2.15}$$

Prin urmare, dacă sistemul restricţiilor conţine $r \leqslant m$ inecuaţii, el poate fi înlocuit printr-un sistem echivalent de tot atîtea ecuaţii, cu $r + n$ variabile; aceste variabile sînt toate supuse condiţiilor de nenegativitate.

O variabilă introdusă într-un program liniar pentru a transforma o inecuaţie într-o ecuaţie este denumită *variabilă complementară*, sau suplimentară.

Dacă pentru a rezolva un program liniar ce conţine inecuaţii am introdus variabile complementare şi am rezolvat apoi problema, ca să obţinem o soluţie optimă a programului liniar iniţial va fi suficient să lăsăm la o parte, din soluţia optimă a programului liniar rezolvat, valorile variabilelor complementare.

Prin urmare, în continuare ne vom ocupa de problema : să se minimizeze funcţia

$$f = c'X, \tag{2.16}$$

pe mulţimea soluţiilor sistemului de condiţii

$$AX = b, \quad X \geqslant 0. \tag{2.17}$$

Scanned with OKEN Scanner

O problemă de programare liniară aflată sub forma (2.16), (2.17) se spune că este o *problemă standard*.

Desigur, denumirea de „liniară" dată problemei considerate se datorește faptului că sistemul restricțiilor este un sistem liniar și funcția obiectiv este o funcție liniară. Dacă, în problema generală a programării liniare, am menține sistemul restricțiilor liniar, dar am înlocui funcția obiectiv printr-un polinom de gradul al doilea în variabilele problemei, am obține problema generală a programării pătratice. Mai general, dacă sistemul restricțiilor sau/și funcția obiectiv nu sînt funcții liniare, problema formulată analog se va numi problemă de programare neliniară.

2.3. PROGRAME LINIARE ECHIVALENTE, EXPLICITAREA UNUI PROGRAM LINIAR

Fie (P) și (P') două probleme de programare liniară de forma standard; să notăm cu S și S' mulțimile de soluții ale lui (P) și (P'); S_a și S'_a — mulțimile de soluții admisibile ale lui (P) și (P'); S_{opt} și S'_{opt} — mulțimile de soluții optime ale lui (P) și (P'). Vom spune că programele liniare (P) și (P') sînt echivalente, dacă

$$S = S', \quad S_a = S'_a, \quad S_{opt} = S'_{opt}. \tag{2.18}$$

Din această definiție rezultă că, plecînd de la un program liniar (P) și aplicînd sistemului său de restricții un șir de transformări elementare, toate programele liniare obținute prin aceste transformări sînt echivalente celui dat. De asemenea, dacă într-un program liniar eliminăm din funcția obiectiv unele variabile cu ajutorul restricțiilor, programul liniar obținut va fi echivalent celui dat.

Vom spune că *programul liniar* (P) este *sub formă explicită* dacă: a) sistemul de restricții al problemei este de formă explicită; b) funcția obiectiv nu conține variabilele principale ale sistemului explicit al restricțiilor. Cu alte cuvinte, programul liniar (P) este sub formă explicită dacă matricea A conține coloanele unei matrice unitate de ordin egal cu numărul ecuațiilor și componentele vectorului c corespunzătoare acestor coloane sînt nule.

Prin definiție, a explicita un program liniar (P) înseamnă a găsi un program liniar de formă explicită echivalent lui (P). În baza rezultatelor din cap. 1 este clar că putem explicita un program liniar dat, dacă și numai dacă sistemul restricțiilor este compatibil,

Scanned with OKEN Scanner

iar în caz de compatibilitate vom putea realiza aceasta prin transformări elementare succesive și eliminarea variabilelor principale ale sistemului explicit din funcția obiectiv, cu ajutorul ecuațiilor sistemului. Desigur, eliminarea unei variabile principale din funcția obiectiv se poate face în același pas cu eliminarea aceleiași variabile din $m - 1$ ecuații ale sistemului; în felul acesta vom obține odată cu o formă explicită a restricțiilor și explicitarea problemei. Remarcăm că prin eliminarea variabilelor principale din funcția obiectiv vor apărea termeni liberi în funcțiile obiectiv obținute succesiv; dealtfel chiar funcția obiectiv din programul liniar dat ar fi putut conține un termen liber. Nu s-a menționat prezența unui asemenea termen în formularea problemei generale datorită faptului că adăugarea sau suprimarea unei constante din funcția obiectiv produce un program liniar echivalent celui dat.

Așadar, pentru a explicita un program liniar dat, sau a stabili incompatibilitatea sistemului restricțiilor, trebuie să aplicăm acestui sistem metoda eliminării complete, dată în § 1.2, completată cu o prevedere care să elimine din funcția obiectiv, în fiecare pas, variabila principală determinată în acel pas. Am văzut că fiecare pas al metodei se încheie cu un pivotaj, având un pivot determinat în cursul pasului. Pivotajul constă în împărțirea la pivot a ecuației care-l conținea, urmată de eliminarea variabilei principale respective din celelalte ecuații. Acum va trebui să facem eliminarea acestei variabile și din funcția obiectiv. Dacă (P^r) este programul liniar obținut după pasul r și $f^{(r)}$ funcția sa obiectiv, anume $f^{(r)} = \alpha_0 + \alpha_1 x_1 + \ldots + \alpha_n x_n$, remarcăm că putem privi funcția obiectiv ca fiind o ecuație scrisă sub forma

$$\alpha_1 x_1 + \ldots + \alpha_n x_n = -\alpha_0 + f^{(r)}, \tag{2.19}$$

cu parametrul $f^{(r)}$ în membrul drept; prin urmare, vom face eliminarea acelei variabile principale și din (2.19) ca din oricare altă ecuație a sistemului restricțiilor. De aceea, la matricea lărgită a sistemului restricțiilor vom adăuga o linie suplimentară, să zicem deasupra matricei, și vom trata această linie în cursul pivotajului ca pe oricare alta. Desigur, nu vom căuta un pivot în linia suplimentară și nu vom uita că termenul liber din linia suplimentară este cel din funcția obiectiv cu semn schimbat. Menționăm că orice matrice de pivotaj de tipul (1.21) va trebui să fie o matrice de ordinul $m + 1$.

Scanned with OKEN Scanner

Toate consideraţiile de mai sus s-ar putea spune mult mai pe scurt sub o altă formă, anume : pentru a explicita un program liniar dat, sau a stabili incompatibilitatea sistemului restricţiilor, vom adăuga la sistemul restricţiilor ecuaţia

$$x_0 + c_1x_1 + \ldots + c_nx_n = 0, \tag{2.20}$$

(luată ca ecuaţia zero) şi vom aplica metoda de explicitare a unui sistem liniar asupra sistemului de $m + 1$ ecuaţii cu n variabile astfel format, adică asupra matricei $\overline{\mathcal{A}} = \begin{pmatrix} 1 & c' \\ 0 & A \end{pmatrix}$.

Într-adevăr, variabila x_0 nou introdusă prin înlocuirea $f = -x_0$ nu apare decît în ecuaţia zero, cu coeficient unitar, deci primul pivotaj este deja realizat şi urmează a se face pivotajele cu pivoţi din celelalte linii, pivotaje care sînt însoţite de eliminarea fiecărei variabile principale şi din linia zero. Diferenţa, neesenţială, este că pe această cale am introduce în plus coloana unitară cu coeficienţii lui x_0.

Am dat mai sus explicaţii detaliate asupra explicitării, deoarece linia corespunzătoare funcţiei obiectiv va juca în continuare un rol deosebit.

Exemplul 2.1. Să considerăm un program liniar (P) : să se minimizeze

$$f = x_1 + x_2 - x_3 - x_4, \tag{2.21}$$

cu restricţiile

$$\left.\begin{aligned} x_1 - 2x_2 + 2x_3 - x_4 &= 2 \\ 2x_1 + x_2 \qquad\quad + x_4 &= 1 \\ 3x_1 + 4x_2 - 2x_3 + 3x_4 &= 0 \end{aligned}\right\} \tag{2.22}$$

şi

$$x_1 \geqslant 0,\ x_2 \geqslant 0,\ x_3 \geqslant 0,\ x_4 \geqslant 0. \tag{2.23}$$

Să explicităm programul liniar (P).

Remarcăm că sistemul restricţiilor programului este sistemul liniar din exemplul 1.2, luat pentru $\lambda = 0$, adică pentru valoarea lui λ ce asigură compatibilitatea ; în felul acesta vom putea compara paşii metodei de explicitare din cele două exemple.

Punem $f = -x_0$ şi scriem (2.21) sub forma

$$x_0 + x_1 + x_2 - x_3 - x_4 = 0, \tag{2.24}$$

ecuație pe care o adăugăm sistemului (2. 22); vom aplica metoda de explicitare a unui sistem liniar asupra matricei

$$\overline{\mathscr{A}} = \left(\begin{array}{ccccc|c} 1 & 1 & 1 & -1 & -1 & 0 \\ \hline 0 & 1 & -2 & 2 & -1 & 2 \\ 0 & 2 & 1 & 0 & 1 & 1 \\ 0 & 3 & 4 & -2 & 3 & 0 \end{array}\right). \tag{2.25}$$

Pasul 1 este deja realizat, sau este realizat prin produsul $M_1\overline{\mathscr{A}}$, unde M_1 este matricea unitate de ordin patru.

Pasul 2. Găsim un pivot în poziția (2, 2) și facem pivotajul formînd cu elementele coloanei a doua vectorul (2; −1, 1, −2, −3), ce reprezintă matricea

$$M_2 = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -2 & 1 & 0 \\ 0 & -3 & 0 & 1 \end{pmatrix}, \tag{2.26}$$

apoi calculăm produsul

$$M_2(M_1\overline{\mathscr{A}}) = \left(\begin{array}{ccccc|c} 1 & 0 & 3 & -3 & 0 & -2 \\ \hline 0 & 1 & -2 & 2 & -1 & 2 \\ 0 & 0 & 5 & -4 & 3 & -3 \\ 0 & 0 & 10 & -8 & 6 & -6 \end{array}\right). \tag{2.27}$$

Pasul 3. Găsim un pivot în poziția (3, 3) și facem pivotajul formînd cu elementele coloanei a treia vectorul (3; − 3/5, 2/5, 1/5, − 2), ce reprezintă matricea

$$M_3 = \begin{pmatrix} 1 & 0 & -3/5 & 0 \\ 0 & 1 & 2/5 & 0 \\ 0 & 0 & 1/5 & 0 \\ 0 & 0 & -2 & 1 \end{pmatrix}, \tag{2.28}$$

apoi calculăm produsul

$$M_3(M_2(M_1\overline{\mathscr{A}})) = \left(\begin{array}{ccccc|c} 1 & 0 & 0 & -3/5 & -9/5 & -1/5 \\ \hline 0 & 1 & 0 & 2/5 & 1/5 & 4/5 \\ 0 & 0 & 1 & -4/5 & 3/5 & -3/5 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right). \tag{2.29}$$

Scanned with OKEN Scanner

Pasul 4. Nu mai găsim un pivot în ultima linie, dar sistemul restricţiilor este compatibil, deci programul liniar este explicitat sub forma : să se minimizeze

$$f^{(4)} = 1/5 - 3/5\ x_3 - 9/5\ x_4, \tag{2.30}$$

cu restricţiile

$$\left.\begin{aligned} x_1 + 2/5\ x_3 + 1/5\ x_4 &= 4/5 \\ x_2 - 4/5\ x_3 + 3/5\ x_4 &= -3/5 \end{aligned}\right\} \tag{2.31}$$

şi

$$x_1 \geqslant 0, \quad x_2 \geqslant 0, \quad x_3 \geqslant 0, \quad x_4 \gtrless 0. \tag{2.32}$$

Programul liniar explicitat a fost scris din matricea (2.29) avînd grijă a scrie termenul liber din $f^{(4)}$ egal cu termenul liber din linia superioară luat cu semn schimbat. Desigur, explicitarea se putea realiza şi prin transformări elementare succesive asupra matricelor (2.25), (2.27).

2.4. SOLUŢII DE BAZĂ ALE UNUI PROGRAM LINIAR

Să presupunem că un program liniar dat (P) poate fi explicitat în raport cu cel puţin un grup de m variabile principale. Atunci ne putem pune problema de a explicita programul (P) în raport cu m variabile date. Pentru a rezolva această problemă vom aplica metoda de explicitare tot asupra matricei $\bar{\mathscr{A}}$, dar nu vom căuta pivoţii succesivi decît în coloanele variabilelor prescrise. Desigur, este posibil ca explicitarea să nu se poată face în raport cu grupul de variabile principale dat dinainte, deşi programul liniar poate fi explicitat faţă de alt grup de m variabile principale.

Fie (P) un program liniar cu m restricţii şi n variabile, despre care presupunem că poate fi explicitat în raport cu cel puţin un grup de m variabile. Vom numi *formă explicită* a programului liniar (P) orice program liniar, echivalent cu (P), aflat sub formă explicită. Din teorema 1.4 rezultă următoarea teoremă analoagă :

Teorema 2.1. *Un program liniar, care admite cel puţin o formă explicită, va admite cel mult* C_n^m *forme explicite, unde* m *este numărul ecuaţiilor şi* n *numărul variabilelor unei forme explicite.*

Dacă (P') este o formă explicită a programului liniar (P), ale cărei variabile principale au indicii într-o mulţime I' şi $I'' = I' - \{i\} + \{k\}$, $i \in I'$, $k \notin I'$, vom putea stabili foarte simplu, fie inexistenţa unei forme explicite a lui (P) cu variabilele principale avînd indicii în I'', fie însăşi această formă explicită, în caz contrar. Anume, dacă elementul din matricea $\bar{\mathscr{A}}'$ a programului (P') situat

în linia corespunzătoare ecuației unde x_i este variabilă principală și în coloana corespunzătoare variabilei x_k este nul, am probat inexistența formei explicite căutate; dacă acest element este nenul și-l luăm drept pivot într-un pivotaj asupra matricei $\bar{\mathcal{A}}'$, atunci după pivotaj vom obține forma explicită căutată. Din cele relatate rezultă că putem trece printr-un pivotaj de la o formă explicită la o altă formă explicită diferind de prima printr-o variabilă principală, atunci cînd a doua există. Desigur, pe această cale vom putea obține lista tuturor formelor explicite ale unui program liniar dat.

Exemplul 2.2. Să considerăm forma explicită (2.30)—(2.32) a programului liniar din exemplul precedent. Matricea $\bar{\mathcal{A}}'$ asociată acestei forme explicite este

$$\bar{\mathcal{A}}' = \left(\begin{array}{ccccc|c} 1 & 0 & 0 & -3/5 & -9/5 & -1/5 \\ \hline 0 & 1 & 0 & 2/5 & 1/5 & 4/5 \\ 0 & 0 & 1 & -4/5 & 3/5 & -3/5 \end{array}\right). \tag{2.33}$$

Putem obține încă patru forme explicite ale programului liniar prin pivotajele:

$$\left(\begin{array}{ccccc|c} 1 & 0 & 0 & -3/5 & -9/5 & -1/5 \\ \hline 0 & 1 & 0 & \textcircled{2/5} & 1/5 & 4/5 \\ 0 & 0 & 1 & -4/5 & 3/5 & -3/5 \end{array}\right) \Rightarrow \left(\begin{array}{ccccc|c} 1 & 3/2 & 0 & 0 & -3/2 & 1 \\ \hline 0 & 5/2 & 0 & 1 & 1/2 & 2 \\ 0 & 2 & 1 & 0 & 1 & 1 \end{array}\right),$$

$$\left(\begin{array}{ccccc|c} 1 & 0 & 0 & -3/5 & -9/5 & -1/5 \\ \hline 0 & 1 & 0 & 2/5 & \textcircled{1/5} & 4/5 \\ 0 & 0 & 1 & -4/5 & 3/5 & -3/5 \end{array}\right) \Rightarrow \left(\begin{array}{ccccc|c} 1 & 9 & 0 & 3 & 0 & 7 \\ \hline 0 & 5 & 0 & 2 & 1 & 4 \\ 0 & -3 & 1 & -2 & 0 & -3 \end{array}\right)$$

$$\left(\begin{array}{ccccc|c} 1 & 0 & 0 & -3/5 & -9/5 & -1/5 \\ \hline 0 & 1 & 0 & 2/5 & 1/5 & 4/5 \\ 0 & 0 & 1 & \textcircled{-4/5} & 3/5 & -3/5 \end{array}\right) \Rightarrow \left(\begin{array}{ccccc|c} 1 & 0 & -3/4 & 0 & -9/4 & 1/4 \\ \hline 0 & 1 & 1/2 & 0 & 1/2 & 1/2 \\ 0 & 0 & -5/4 & 1 & -3/4 & 3/4 \end{array}\right),$$

$$\left(\begin{array}{ccccc|c} 1 & 0 & 0 & -3/5 & -9/5 & -1/5 \\ \hline 0 & 1 & 0 & 2/5 & 1/5 & 4/5 \\ 0 & 0 & 1 & -4/5 & \textcircled{3/5} & -3/5 \end{array}\right) \Rightarrow \left(\begin{array}{ccccc|c} 1 & 0 & 3 & -3 & 0 & -2 \\ \hline 0 & 1 & -1/3 & 2/3 & 0 & 1 \\ 0 & 0 & 5/3 & -4/3 & 1 & -1 \end{array}\right).$$

Scanned with OKEN Scanner

Aceste forme explicite sînt respectiv:

$$\left.\begin{aligned} \frac{5}{2}x_1 \qquad + x_3 + \frac{1}{2}x_4 &= 2 \\ 2x_1 + x_2 \qquad + \quad x_4 &= 1 \end{aligned}\right\} \qquad x_i \geqslant 0, \quad (i = 1, 2, 3, 4), \qquad (2.34)$$

$$\frac{3}{2}x_1 - \frac{3}{2}x_4 - 1 = \min;$$

$$\left.\begin{aligned} 5x_1 \qquad + 2x_3 + x_4 &= 4 \\ -3x_1 + x_2 - 2x_3 \qquad &= -3 \end{aligned}\right\} \qquad x_i \geqslant 0, \quad (i = 1, 2, 3, 4), \qquad (2.35)$$

$$9x_1 + 3x_3 - 7 = \min;$$

$$\left.\begin{aligned} x_1 + \frac{1}{2}x_2 \qquad + \frac{1}{2}x_4 &= \frac{1}{2} \\ -\frac{5}{4}x_2 + x_3 - \frac{3}{4}x_4 &= \frac{3}{4} \end{aligned}\right\} \qquad x_i \geqslant 0, \quad (i = 1, 2, 3, 4), \qquad (2.36)$$

$$-\frac{3}{4}x_2 - \frac{9}{4}x_4 - \frac{1}{4} = \min;$$

$$\left.\begin{aligned} x_1 - \frac{1}{3}x_2 + \frac{2}{3}x_3 \qquad &= 1 \\ \frac{5}{3}x_2 - \frac{4}{3}x_3 + x_4 &= -1 \end{aligned}\right\} \qquad x_i \geqslant 0, \quad (i = 1, 2, 3, 4), \qquad (2.37)$$

$$3x_2 - 3x_3 + 2 = \min.$$

Utilizînd una dintre formele explicite nou găsite, vom putea obține a șasea formă explicită a programului liniar, anume

$$\left.\begin{aligned} \frac{3}{2}x_1 - \frac{1}{2}x_2 + x_3 \qquad &= \frac{3}{2} \\ 2x_1 + \quad x_2 \qquad + x_4 &= 1 \end{aligned}\right\} \qquad x_i \geqslant 0, \quad (i = 1, 2, 3, 4), \qquad (2.38)$$

$$\frac{9}{2}x_1 + \frac{3}{2}x_2 - \frac{5}{2} = \min.$$

Scanned with OKEN Scanner

Orice soluţie de bază a sistemului de restricţii dintr-un program liniar se numeşte *soluţie de bază* a programului liniar.

Din această definiţie rezultă că fiecărei forme explicite a programului liniar îi corespunde o soluţie de bază; valoarea funcţiei obiectiv a programului, pentru soluţia de bază asociată formei explicite, este termenul liber din funcţia obiectiv a formei explicite. Pentru a trece de la o soluţie de bază la alta, diferind de prima printr-o variabilă principală, putem trece de la forma explicită căreia îi este asociată prima soluţie de bază la forma explicită asociată celei de-a doua şi apoi să citim noua soluţie de bază, împreună cu valoarea funcţiei obiectiv, în ultima formă explicită,

Exemplul 2.3. Programul liniar (2.21) — (2.23) are şase soluţii de bază, corespunzătoare celor şase forme explicite ale sale. Soluţia de bază corespunzătoare formei explicite (2.30) — (2.32) şi valoarea funcţiei obiectiv pentru această soluţie sînt

$$x_1 = 4/5,\ x_2 = -3/5,\ x_3 = 0,\ x_4 = 0,\ f = 1/5. \tag{2.39}$$

Formele explicite (2.34) — (2.38) dau următoarele soluţii de bază şi valori corespunzătoare ale funcţiei obiectiv:

$$(0,\ 1,\ 2,\ 0;\ -1),\quad (0,\ -3,\ 0,\ 4;\ -7), \tag{2.40}$$

$$(1/2,\ 0,\ 3/4,\ 0;\ -1/4),\ (1,\ 0,\ 0,\ -1;\ 2),\ (0,\ 0,\ 3/2,\ 1;\ -5/2).$$

Trecerile de la o soluţie de bază la alta sînt pivotajele din exemplul precedent.

Se observă că prima coloană din matricele cu care s-a lucrat în cele două exemple precedente, coloana asociată variabilei x_0, este inutilă; de aceea ea va fi omisă în cele ce urmează.

2.5. PIVOTAJ PRIMAL ASUPRA UNUI PROGRAM LINIAR

Fie (P) un program liniar sub formă explicită. Cu alte cuvinte, programul (P) satisface condiţiile: a) matricea sistemului conţine toate coloanele matricei unitate de acelaşi ordin cu numărul ecuaţiilor sistemului; b) variabilele principale ale sistemului restricţiilor nu intră în funcţia obiectiv.

Să notăm cu I mulţimea indicilor variabilelor principale şi K mulţimea indicilor variabilelor secundare din sistemul restricţiilor. Atunci programul liniar (P) se formulează astfel: să se minimizeze

$$f = f_0 + \sum_{k \in K} \alpha_0^k x_k, \tag{2.41}$$

cu restricţiile

$$x_i + \sum_{k \in K} \alpha_i^k x_k = \beta_i, \qquad (i \in I), \tag{2.42}$$

şi

$$x_i \geqslant 0, \; x_k \geqslant 0, \qquad (i \in I, \quad k \in K). \tag{2.43}$$

Vom spune că programul liniar (P) este sub o *formă explicită* primal admisibilă dacă, pe lîngă condiţiile a) şi b), el satisface şi condiţia : c) termenii liberi ai sistemului restricţiilor sînt nenegativi, deci $\beta_i \geqslant 0$ pentru toţi $i \in I$.

Remarcăm că dacă programul liniar (P) este sub o formă explicită primal admisibilă, atunci soluţia de bază corespunzătoare formei explicite a programului

$$x_i = \beta_i, \; x_k = 0, \; (i \in I, \; k \in K), \tag{2.44}$$

este o soluţie de bază admisibilă, valoarea funcţiei obiectiv pentru această soluţie este $f = f_0$.

Aşa cum arătam în paragraful precedent, programului liniar (P) i se asociază o matrice $\bar{\mathscr{A}}$ formată dintr-o linie ce cuprinde coficienţii şi termenul liber din funcţia obiectiv şi liniile matricei lărgite a sistemului restricţiilor.

Să demonstrăm teorema următoare :

Teorema 2.2. *Dacă (P) este un program liniar sub o formă explicită primal admisibilă şi perechea de indici $j \in I$ şi $h \in K$ este astfel încî*

$$(\alpha) \quad \alpha_0^h < 0, \qquad (\beta) \; \frac{\beta_j}{\alpha_j^h} = \min_{i \mid \alpha_i^h > 0} \frac{\beta_i}{\alpha_i^h},$$

atunci un pivotaj cu elementul pivot α_j^h efectuat asupra matricei $\bar{\mathscr{A}}$ va conduce la o formă explicită tot primal admisibilă şi valoarea funcţiei obiectiv pentru soluţia de bază corespunzătoare este mai mică sau cel mult egală cu valoarea obţinută pentru soluţia de bază de plecare.

Demonstraţie. Să efectuăm un pivotaj asupra matricei problemei $\bar{\mathscr{A}}$ punînd în evidenţă în matrice numai elementele ce prezintă interes mai departe. Atragem atenţia că un element α_i^k din matrice nu are ambii indici cu semnificaţia obişnuită, ci indicele de jos este indicele variabilei principale din linia unde se găseşte elemen-

tul, în timp ce indicele de sus arată, ca de obicei, coloana unde se găsește elementul. Cu această precizare, pivotajul anunțat în teoremă este următorul:

$$\bar{\mathscr{A}} = \left(\begin{array}{c|c} \cdots\ \alpha_0^h\ \cdots & -f_0 \\ \hline \vdots & \vdots \\ \cdots\ \alpha_j^h\ \cdots & \beta_j \\ \vdots & \vdots \\ \cdots\ \alpha_i^h\ \cdots & \beta_i \\ \vdots & \vdots \\ \downarrow & \end{array}\right)$$

$$\bar{\mathscr{A}}' = \left(\begin{array}{c|c} \downarrow & \\ \cdots\ 0\ \cdots & -\left(f_0 + \frac{\beta_j}{\alpha_j^h}\alpha_0^h\right) \\ \vdots & \vdots \\ \cdots\ 1\ \cdots & \frac{\beta_j}{\alpha_j^h} \\ \vdots & \vdots \\ \cdots\ 0\ \cdots & \beta_i - \frac{\beta_j}{\alpha_j^h}\alpha_i^h \\ \vdots & \vdots \end{array}\right)$$

Este cert că după pivotaj obținem o formă explicită nouă a programului (P); va trebui să dovedim numai admisibilitatea primală a acestei forme. Noua soluție de bază este

$$\begin{aligned} &\bar{x}_h = \frac{\beta_j}{\alpha_j^h}, \quad \bar{x}_i = \beta_i - \frac{\beta_j}{\alpha_j^h}\alpha_i^h, \qquad (i \in I,\ i \neq j), \\ &\bar{x}_j = 0, \qquad \bar{x}_k = 0, \qquad\qquad (k \in K,\ k \neq h), \end{aligned} \tag{2.45}$$

și dă valoarea funcției obiectiv

$$\bar{f} = f_0 + \frac{\beta_j}{\alpha_j^h}\alpha_0^h. \tag{2.46}$$

Pentru a stabili admisibilitatea, dovedim admisibilitatea soluției de bază (2.45). Ipoteza (β) asigură $\alpha_j^h > 0$, deoarece minimul din

membrul drept este luat numai pentru rapoartele cu numitori pozitivi; condiția de admisibilitate a formei de plecare era $\beta_i \geqslant 0$, $(i \in I)$, deci și $\beta_j \geqslant 0$; rezultă $\bar{x}_h \geqslant 0$. Apoi, putem scrie în (2.45) componentele $\bar{x}_i = \beta_i - \bar{x}_h \alpha_i^h$, $(i \in I,\ i \neq j)$, de unde este clar că pentru acei i pentru care $\alpha_i^h \leqslant 0$ vom avea $\beta_i \geqslant 0$ din condiția de admisibilitate și cum $\bar{x}_h \geqslant 0$ (deja s-a demonstrat), rezultă $\bar{x}_i \geqslant 0$. Pe de altă parte, pentru un indice i cu $\alpha_i^h > 0$, din (β) găsim

$$\frac{\beta_j}{\alpha_j^h} \leqslant \frac{\beta_i}{\alpha_i^h}; \tag{2.47}$$

putem scrie

$$\bar{x}_i = \alpha_i^h \left(\frac{\beta_i}{\alpha_i^h} - \frac{\beta_j}{\alpha_j^h} \right), \tag{2.48}$$

de unde $\bar{x}_i \geqslant 0$.

Valoarea funcției obiectiv pentru soluția de bază (2.45) este dată de formula (2.46); cum avem din demonstrația de mai sus $\frac{\beta_j}{\alpha_j^h} \geqslant 0$, iar $\alpha_0^h < 0$ din condiția (α), va rezulta $\bar{f} \leqslant f_0$.

Demonstrația teoremei 2.2 permite formularea cîtorva observații.

a) Condiția (β), independent de condiția (α), asigură admisibilitatea primală a noii forme explicite a programului liniar. Mai observăm că această condiție presupune existența în coloana h, ce satisface (α), a cel puțin unui element $\alpha_i^h > 0$, altfel condiția (β) rămîne fără obiect, căci însăși transformarea este nedeterminată, fiind nedeterminat pivotul α_j^h. Presupunînd existența a cel puțin unui astfel de element, prin condiția (β) indicele j va fi, în general, unic determinat; desigur, aceasta nu se va întîmpla cînd există mai multe rapoarte cu valoarea minimă.

b) Condiția (α) presupune existența în linia zero a cel puțin unui element $\alpha_0^k < 0$. Presupunînd existența a cel puțin unui astfel de element, prin condiția (α) indicele h, în general, nu este unic determinat. Acest fapt devine un inconvenient la formularea unui algoritm care să permită trecerea automată de la o soluție de bază admisibilă la alta. De aceea, în loc de a se considera dintre indicii ce satisfac condiția (α) unul oarecare, se va considera în continuare indicele h determinat prin regula

$$\alpha_0^h = \min\ \alpha_0^k, \tag{2.49}$$

Scanned with OKEN Scanner

indice, care, în general, este unic; desigur, aceasta nu se va întîmpla cînd există mai mulţi coeficienţi α_0^k cu valoarea minimă. Bineînţeles, dacă h este determinat prin (2.49), nu rezultă că el satisface condiţia (α).

c) Condiţia (α) ne asigură că valoarea funcţiei obiectiv nu va creşte prin trecerea la o nouă soluţie de bază admisibilă. Mai mult, dacă $\beta_j > 0$, vom avea chiar $\bar{f} < f_0$; de exemplu, dacă soluţia de bază admisibilă de plecare este nedegenerată, vom obţine scăderea valorii funcţiei obiectiv. Se înţelege, presupunerea nondegenerării soluţiei de bază de plecare (2.44) este o condiţie suficientă pentru scăderea valorii funcţiei obiectiv, fără a fi şi necesară.

Din teorema 2.2 şi observaţiile ulterioare rezultă:

Teorema 2.3. *Dacă (P) este un program liniar explicit primal admisibil şi perechea de indici $j \in I$, $h \in K$, este determinată prin regulile*

$$\alpha_0^h = \min_k \alpha_0^k, \quad \frac{\beta_j}{\alpha_j^h} = \min_{i \mid \alpha_i^h > 0} \frac{\beta_i}{\alpha_i^h}, \tag{2.50}$$

iar $\alpha_0^h < 0$, atunci pivotajul cu elementul pivot α_j^h conduce la o nouă formă explicită tot admisibilă şi, dacă soluţia de bază de plecare era nedegenerată, valoarea funcţiei obiectiv pentru soluţia de bază nouă este mai mică decît valoarea funcţiei obiectiv pentru soluţia de bază de plecare.

Teorema 2.3 cuprinde de fapt un algoritm prin care plecînd de la o formă explicită primal admisibilă a unui program liniar vom obţine automat o nouă formă explicită primal admisibilă. Este necesară doar precizarea suplimentară, că în cazul neunicităţii cel puţin unuia dintre minimile din regulile (2.50) se va alege primul indice h sau respectiv j, care le satisface. Lăsînd la o parte deocamdată formularea algoritmului, vom face două observaţii foarte importante.

Algoritmul cuprins în teorema 2.3 permite construirea de noi şi noi forme explicite primal admisibile atîta vreme cît nu întîlnim vreuna dintre următoarele două situaţii:

(I) $\alpha_0^k \geqslant 0$ pentru orice $k \in K$;

(II) $\alpha_0^h = \min_k \alpha_0^k < 0$ şi $\alpha_i^h \leqslant 0$ pentru orice $i \in I$.

Într-adevăr, în cazul (I) nu vom putea determina indicele h, (pentru care $\alpha_0^h < 0$), iar în cazul (II) nu vom putea determina indicele j.

Dacă toate soluţiile de bază corespunzătoare formelor explicite primal admisibile ale programului liniar construite pe baza algoritmului sînt nedegenerate, este sigur că orice nouă formă explicită construită va fi diferită de cele precedente, deoarece valoarea funcţiei obiectiv scade. Conform teoremei 2.1 există numai un număr finit de forme explicite ale unui program liniar, de unde rezultă că : dacă toate soluţiile de bază construite în timpul aplicării algoritmului sînt nedegenerate, atunci după un număr finit de paşi vom obţine una dintre situaţiile (I) sau (II).

Vom denumi *pivotaj primal* asupra unui program liniar de formă explicită primal admisibilă un pivotaj în care pivotul α_j^h este determinat prin regulile (2.50). Prima regulă determină indicele $h \in K$ al variabilei x_h care „intră'' între cele principale, iar a doua determină indicele $j \in I$ al variabilei x_j care „iese'' dintre cele principale. De aceea prima regulă este denumită *criteriul de intrare*, iar a doua regulă este denumită *criteriul de ieşire* al pivotajului primal.

Exemplul 2.4. Să considerăm programul liniar (P) : să se minimizeze

$$h = 5 - 2w_1 - 3w_2, \tag{2.51}$$

cu restricţiile

$$\left.\begin{aligned} u_1 \quad + w_1 - w_2 &= 1 \\ u_2 \quad - w_1 + w_2 &= 2 \\ u_3 + 2w_1 + w_2 &= 3 \end{aligned}\right\} \tag{2.52}$$

şi

$$u_1 \geqslant 0, \quad u_2 \geqslant 0, \quad u_3 \geqslant 0, \quad w_1 \geqslant 0, \quad w_2 \geqslant 0. \tag{2.53}$$

Programul liniar (P) este sub formă explicită primal admisibilă şi soluţia de bază corespunzătoare este

$$u_1 = 1, \quad u_2 = 2, \quad u_3 = 3, \quad w_1 = 0, \quad w_2 = 0, \tag{2.54}$$

iar valoarea funcţiei obiectiv pentru această soluţie este $h = 5$.

Ne propunem să trecem, folosind algoritmul cuprins în regulile (2.50), de la o formă explicită primal admisibilă la alta, pînă cînd vom întîlni una dintre situaţiile (I) sau (II). Totodată, vom scrie şi soluţiile de bază corespunzătoare formelor explicite întîlnite. Matricea $\overline{\mathscr{A}}$ a formei explicite de plecare este

$$\overline{\mathscr{A}} = \left(\begin{array}{ccccc:c} 0 & 0 & 0 & -2 & \overset{\downarrow}{-3} & -5 \\ \hdashline 1 & 0 & 0 & 1 & -1 & 1 \\ 0 & 1 & 0 & -1 & \textcircled{1} & 2 \\ 0 & 0 & 1 & 2 & 1 & 3 \end{array}\right) \rightarrow .$$

Criteriul de intrare determină variabila w_2 ce devine principală, iar criteriul de ieșire determină variabila u_2 ce devine secundară; pivotajul primal conduce la matricea

$$\overline{\mathscr{A}}^1 = \begin{pmatrix} 0 & 3 & 0 & -5 & 0 & 1 \\ \hline 1 & 1 & 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & -1 & 1 & 2 \\ 0 & -1 & 1 & \textcircled{3} & 0 & 1 \end{pmatrix} \rightarrow,$$

iar soluția de bază corespunzătoare este

$$u_1^1 = 3, \quad u_2^1 = 0, \quad u_3^1 = 1, \quad w_1^1 = 0, \quad w_2^1 = 2, \tag{2.55}$$

pentru care $h^1 = -1$. Criteriul de intrare determină variabila w_1 ce devine principală, iar criteriul de ieșire determină variabila u_3 ce devine secundară; noul pivotaj primal conduce la matricea

$$\overline{\mathscr{A}}^2 = \begin{pmatrix} 0 & 4/3 & 5/3 & 0 & 0 & 8/3 \\ \hline 1 & 1 & 0 & 0 & 0 & 3 \\ 0 & 2/3 & 1/3 & 0 & 1 & 7/3 \\ 0 & -1/3 & 1/3 & 1 & 0 & 1/3 \end{pmatrix},$$

iar soluția de bază corespunzătoare este

$$u_1^2 = 3, \quad u_2^2 = 0, \quad u_3^2 = 0, \quad w_1^2 = 1/3, \quad w_2^2 = 7/3, \tag{2.56}$$

pentru care $h^2 = -8/3$. Matricea $\overline{\mathscr{A}}^2$ prezintă situația (I), deci nu vom mai putea aplica un nou pivotaj primal.

Este ușor de văzut cum au fost determinați pivoții din matricele $\mathscr{A}$ și $\overline{\mathscr{A}}^1$, pe baza regulilor (2.50); pivotajele au fost realizate prin transformări elementare, cu toate că pot fi realizate și cu ajutorul matricelor de pivotaj.

2.6. PIVOTAJ DUAL ASUPRA UNUI PROGRAM LINIAR

Fie (P) un program liniar sub o formă explicită. Să întrebuințăm pentru formularea lui notațiile din paragraful precedent, adică să presupunem că el se scrie sub forma (2.41) — (2.43).

Vom spune că programul liniar (P) este sub o formă explicită dual admisibilă dacă, pe lîngă condițiile a), b), satisface și condiția: d) toți coeficienții necunoscutelor din funcția obiectiv sînt nenegativi; deci $\alpha_0^k \geqslant 0$ pentru toți $k \in K$. Soluția de bază corespunzătoare

unei forme explicite dual admisibile a unui program liniar se va numi *soluție de bază dual admisibilă*. Remarcăm că o soluție de bază dual admisibilă nu este, în general, soluție de bază primal admisibilă a programului liniar.

Fie $\bar{\mathscr{A}}$ matricea asociată programului liniar (P); să demonstrăm teorema următoare:

Teorema 2.4. *Dacă (P) este un program liniar sub o formă explicită dual admisibilă și perechea de indici $j \in I$ și $h \in K$ este astfel încît*

$$(\gamma) \quad \beta_j < 0, \qquad (\delta) \quad \frac{\alpha_0^h}{\alpha_j^h} = \max_{k \mid \alpha_j^k < 0} \frac{\alpha_0^k}{\alpha_j^k},$$

atunci un pivotaj cu elementul pivot α_j^h, efectuat asupra matricei $\bar{\mathscr{A}}$, va conduce la o formă explicită tot dual admisibilă și valoare a funcției obiectiv pentru soluția de bază corespunzătoare este mai mare sau cel puțin egală cu valoarea obținută pentru soluția de plecare.

Demonstrație. Să efectuăm pivotajul asupra matricei problemei $\bar{\mathscr{A}}$, punînd în evidență în matrice numai elementele ce prezintă interes mai departe. Atragem atenția că semnificația indicilor unui element din matrice, întrucîtva diferită de cea obișnuită, este cea din paragraful precedent. Cu această precizare, pivotajul anunțat în teoremă este următorul:

$$\bar{\mathscr{A}} = \left(\begin{array}{ccccc|c} \dots & 0 & \dots & \alpha_0^h & \dots \; \alpha_0^k \; \dots & -f_0 \\ \hline \vdots & \vdots & & \vdots & & \vdots \\ \dots & 1 & \dots & \alpha_j^h & \dots \; \alpha_j^k \; \dots & \beta_j \\ \vdots & \vdots & & \vdots & & \vdots \end{array}\right)$$

$$\Downarrow$$

$$\bar{\mathscr{A}} = \left(\begin{array}{ccccc|c} \dots & -\dfrac{\alpha_0^h}{\alpha_j^h} & \dots & 0 & \dots \; \alpha_0^k - \dfrac{\alpha_j^k}{\alpha_j^h} \cdot \alpha_0^h \; \dots & -\left(f_0 + \dfrac{\beta_j}{\alpha_j^h} \cdot \alpha_0^h\right) \\ \hline \vdots & \vdots & & \vdots & & \vdots \\ \dots & \dfrac{1}{\alpha_j^h} & \dots & 1 & \dots \quad \dfrac{\alpha_j^k}{\alpha_j^h} \quad \dots & \dfrac{\beta_j}{\alpha_j^h} \\ \vdots & \vdots & & \vdots & & \vdots \end{array}\right).$$

Este cert că după pivotaj se obţine o formă explicită nouă a programului (P); va trebui să dovedim numai admisibilitatea duală a acestei forme. Coeficienţii noi ai funcţiei obiectiv sînt

$$\bar{\alpha}_0^j = -\frac{\alpha_0^h}{\alpha_j^h}, \quad \bar{\alpha}_0^k = \alpha_0^k - \frac{\alpha_0^h}{\alpha_j^h}\alpha_j^k, \qquad (k \in K,\ k \neq h)$$
$$\bar{\alpha}_0^h = 0, \qquad \bar{\alpha}_0^i = 0, \qquad (i \in I,\ i \neq j), \tag{2.57}$$

iar valoarea funcţiei obiectiv pentru soluţia de bază corespunzătoare formei explicite este

$$\bar{f} = f_0 + \frac{\alpha_0^h}{\alpha_j^h}\beta_j. \tag{2.58}$$

Pentru a stabili admisibilitatea duală, dovedim nenegativitatea coeficienţilor (2.57). Ipoteza (δ) asigură $\alpha_j^h < 0$, deoarece maximul din membrul drept este luat numai pentru rapoartele cu numitori negativi; condiţia de admisibilitate duală a formei de plecare era $\alpha_0^k \geqslant 0$, $(k \in K)$, deci şi $\alpha_0^h \geqslant 0$; rezultă $\bar{\alpha}_0^j \geqslant 0$. Apoi, putem scrie în (2.57) coeficienţii $\bar{\alpha}_0^k = \alpha_0^k + \bar{\alpha}_0^j \alpha_j^k$, $(k \in K,\ k \neq j)$, de unde este clar că pentru acei k pentru care $\alpha_j^k \geqslant 0$ vom avea $\alpha_0^k \geqslant 0$ din condiţia de admisibilitate duală, $\bar{\alpha}_0^j \geqslant 0$ deja demonstrată şi deci $\bar{\alpha}_0^k \geqslant 0$. Acum pentru un indice k astfel ca $\alpha_j^k < 0$, din (δ) găsim

$$\frac{\alpha_0^h}{\alpha_j^h} \geqslant \frac{\alpha_0^k}{\alpha_j^k}, \tag{2.59}$$

şi putem scrie

$$\bar{\alpha}_0^k = -\alpha_j^k\left(\frac{\alpha_0^h}{\alpha_j^h} - \frac{\alpha_0^k}{\alpha_j^k}\right). \tag{2.60}$$

de unde $\bar{\alpha}_0^k \geqslant 0$.

Valoarea funcţiei obiectiv pentru soluţia de bază (2.45) este dată de formula (2.58); cum avem din demonstraţia de mai sus $\alpha_0^h/\alpha_j^h \leqslant 0$, iar $\beta_j < 0$ din condiţia (γ) va rezulta $\bar{f} \geqslant f_0$.

Demonstraţia teoremei 2.4 permite formularea cîtorva observaţii:

a) Condiţia (δ), independent de condiţia (γ), asigură admisibilitatea duală a noii forme explicite a programului liniar. Mai obser-

văm că această condiție presupune existența în linia j, ce satisface (γ), a cel puțin unui element $\alpha_j^k < 0$, altfel condiția (δ) rămîne fără obiect, căci însăși transformarea este nedeterminată, fiind nedeterminat pivotul α_j^h. Presupunînd existența a cel puțin unui asemenea element, prin condiția (δ) indicile h va fi, în general, unic determinat; desigur, aceasta nu se va întîmpla cînd există mai multe rapoarte cu valoarea minimă.

b) Condiția (γ) presupune existența în coloana termenilor liberi a cel puțin unui element $\beta_i < 0$. Presupunînd existența a cel puțin unui astfel de element, prin condiția (γ) indicele j, în general, nu este unic determinat. Acest fapt devine un inconvenient la formularea unui algoritm care să permită trecerea automată de la o soluție de bază dual admisibilă la alta. De aceea, în loc de a se considera dintre indicii ce satisfac condiția (γ) unul oarecare, se va considera în continuare indicele j determinat prin regula

$$\beta_j = \min_i \beta_i, \tag{2.61}$$

indice care, în general, este unic; desigur, aceasta nu se va întîmpla cînd există mai mulți termeni liberi β_i cu valoarea minimă. Bineînțeles, dacă j este determinat prin (2.61), nu rezultă că el satisface condiția (γ).

c) Condiția (γ) ne asigură că valoarea funcției obiectiv nu va scădea prin trecerea la o nouă soluție de bază dual admisibilă. Mai mult, dacă $\alpha_0^h > 0$ vom avea chiar $\bar{f} > f_0$; de exemplu, dacă $\alpha_0^k > 0$ pentru orice $k \in K$ vom obține creșterea valorii funcției obiectiv. Prin analogie, vom spune că soluția de bază dual admisibilă (2.44) este nedegenerată dual, cînd $\alpha_0^k > 0$ pentru orice $k \in K$. Se înțelege, presupunerea nondegenerării duale a soluției de bază de plecare (2.44) este o condiție suficientă pentru creșterea valorii funcției obiectiv, fără a fi și necesară.

Din teorema 2.4 și observațiile ulterioare rezultă:

Teorema 2.5. *Dacă (P) este un program liniar sub o formă explicită dual admisibilă și perechea de indici $j \in I$ și $h \in K$ este determinată prin regulile*

$$\beta_j = \min_i \beta_i, \quad \frac{\alpha_0^h}{\alpha_j^h} = \min_{k|\alpha_j^k<0} \left| \frac{\alpha_0^k}{\alpha_j^k} \right|, \tag{2.62}$$

iar $\beta_j < 0$, *atunci pivotajul cu elementul pivot* α_j^h *conduce la o nouă formă explicită tot dual admisibilă şi, dacă soluţia de bază de plecare era nedegenerată dual, valoarea funcţiei obiectiv pentru soluţia de bază nouă este mai mare decît valoarea funcţiei obiectiv pentru soluţia de bază de plecare.*

Teorema 2.5 cuprinde de fapt un algoritm prin care plecînd de la o formă explicită dual admisibilă a unui program liniar vom obţine automat o nouă formă explicită dual admisibilă. Este necesară doar precizarea suplimentară, că în cazul neunicităţii cel puţin unuia dintre minimele din regulile (2.62) se va alege primul indice h, sau respectiv j, care le satisface. Lăsînd la o parte deocamdată formularea algoritmului, vom face două observaţii foarte importante.

Algoritmul cuprins în teorema 2.5 permite construirea de noi şi noi forme explicite dual admisibile atîta vreme cît nu întîlnim vreuna dintre următoarele două situaţii:

(I') $\beta_i \geqslant 0$ pentru orice $i \in I$;

(II') $\beta_j = \min_i \beta_i < 0$ şi $\alpha_j^k \geqslant 0$ pentru orice $k \in K$.

Într-adevăr, în cazul (I') nu vom putea determina indicele j (pentru care $\beta_j < 0$), iar în cazul (II') nu vom putea determina indicele h.

Dacă toate soluţiile de bază corespunzătoare formelor explicite dual admisibile ale programului liniar construite pe baza algoritmului sînt nedegenerate dual, este sigur că orice nouă formă explicită construită va fi diferită de cele precedente, deoarece valoarea funcţiei obiectiv creşte. Conform teoremei 2.1 există numai un număr finit de forme explicite ale unui program liniar, de unde rezultă că: dacă toate soluţiile de bază construite în timpul aplicării algoritmului sînt nedegenerate dual, atunci după un număr finit de paşi vom obţine una dintre situaţiile (I') sau (II').

Vom denumi *pivotaj dual* asupra unui program liniar de formă explicită dual admisibilă, un pivotaj în care pivotul α_j^h este determinat prin regulile (2.62). Prima regulă determină indicele $j \in I$ al variabilei x_j care „iese" dintre variabilele principale, iar a doua determină indicele $h \in K$ al variabilei x_h care „intră" între cele principale. De aceea, prima regulă este denumită *criteriul de ieşire*, iar a doua regulă *criteriul de intrare* al pivotajului dual.

Exemplul 2.5. Să considerăm programul liniar (P): să se minimizeze

$$f = -5 + x_1 + 2x_2 + 3x_3, \tag{2.63}$$

cu restricţiile

$$\left.\begin{aligned} -x_1 + x_2 - 2x_3 + y_1 \qquad &= -2 \\ x_1 - x_2 - x_3 \qquad + y_2 &= -3 \end{aligned}\right\} \tag{2.64}$$

şi

$$x_1 \geqslant 0, \quad x_2 \geqslant 0, \quad x_3 \geqslant 0, \quad y_1 \geqslant 0, \quad y_2 \geqslant 0. \tag{2.65}$$

Programul liniar (P) este sub formă explicită dual admisibilă şi soluţia de bază dual admisibilă corespunzătoare este

$$x_1 = 0, \quad x_2 = 0, \quad x_3 = 0, \quad y_1 = -2, \quad y_2 = -3, \tag{2.66}$$

iar valoarea funcţiei obiectiv pentru această soluţie este $f = -5$.

Ne propunem să trecem, folosind algoritmul cuprins în regulile (2.62), de la o formă explicită dual admisibilă la alta, pînă cînd vom întîlni una dintre situaţiile (I′) sau (II′). Totodată vom scrie şi soluţiile de baza corespunzătoare formelor explicite întîlnite. Matricea $\overline{\mathscr{B}}$ supusă pivotajului dual este

$$\overline{\mathscr{B}} = \begin{array}{c} \begin{array}{ccccc} & \downarrow & & & \end{array} \\ \left(\begin{array}{ccccc|c} 1 & 2 & 3 & 0 & 0 & 5 \\ \hline -1 & 1 & -2 & 1 & 0 & -2 \\ 1 & -1 & -1 & 0 & 1 & -3 \end{array}\right) \end{array} \begin{array}{c} \\ \\ \cdot\cdot \\ \rightarrow \end{array}$$

Criteriul de ieşire determină varabila y_2 ce devine secundară, iar criteriul de intrare determină variabila x_2 ce devine principală : pivotajul dual conduce la matricea

$$\overline{\mathscr{B}}^1 = \begin{array}{c} \begin{array}{ccccc} & & \downarrow & & \end{array} \\ \left(\begin{array}{ccccc|c} 3 & 0 & 1 & 0 & 2 & -1 \\ \hline 0 & 0 & -3 & 1 & 1 & -5 \\ -1 & 1 & 1 & 0 & -1 & 3 \end{array}\right) \end{array} \begin{array}{c} \\ \\ , \\ \rightarrow \end{array}$$

iar soluţia de bază corespunzătoare este

$$x_1^1 = 0, \quad x_2^1 = 3, \quad x_3^1 = 0, \quad y_1^1 = -5, \quad y_2^1 = 0, \tag{2.67}$$

pentru care $f^1 = 1$. Acum, criteriul de ieşire determină variabila y_1 ce devine secundară, iar criteriul de intrare determină variabila x_3 ce devine principală; noul pivotaj dual conduce la matricea

$$\overline{\mathscr{B}}^2 = \left(\begin{array}{ccccc|c} 3 & 0 & 0 & 1/3 & 7/3 & -8/3 \\ \hline 0 & 0 & 1 & -1/3 & -1/3 & 5/3 \\ -1 & 1 & 0 & 1/3 & -2/3 & 4/3 \end{array}\right),$$

iar soluţia de bază corespunzătoare este

$$x_1^2 = 0, \quad x_2^2 = 4/3, \quad x_3^2 = 5/3, \quad y_1^2 = 0, \quad y_2^2 = 0, \tag{2.68}$$

pentru care $f^2 = 8/3$. Matricea $\overline{\mathscr{B}}^2$ prezintă situaţia (I'), deci nu vom mai putea aplica un nou pivotaj dual.

3

METODE GENERALE DE REZOLVARE A PROGRAMELOR LINIARE: METODA SIMPLEX PRIMALĂ ŞI METODA SIMPLEX DUALĂ

3.1. TEOREME FUNDAMENTALE

Fie (P) un program liniar cu n variabile, care admite o formă explicită. Dacă notăm cu I mulţimea indicilor variabilelor principale şi cu K mulţimea indicilor variabilelor secundare dintr-o formă explicită a programului liniar, atunci programul sub acea formă explicită se scrie: să se minimizeze

$$f = f_0 + \sum_{k \in K} \alpha_0^k x_k \tag{3.1}$$

cu restricţiile

$$x_i + \sum_{k \in K} \alpha_i^k x_k = \beta_i, \ (i \in I), \tag{3.2}$$

şi

$$x_i \geqslant 0, \ x_k \geqslant 0, \ (i \in I, \ k \in K). \tag{3.3}$$

Metodele generale de rezolvare a programelor liniare de care ne vom ocupa în continuare se bazează pe următoarea teoremă:

Teorema 3.1. *Dacă un program liniar are o soluţie optimă, atunci are şi o soluţie optimă de bază.*

Demonstraţie. Justificarea teoremei se bazează pe consideraţii constructive; fiind dată o soluţie optimă, vom arăta cum se poate construi o soluţie optimă de bază. Aşadar, fie $\hat{x} = (\hat{x}_j)$ o soluţie optimă a programului liniar (P) şi $\hat{f}$ valoarea corespunzătoare a funcţiei obiectiv. Fie s numărul componentelor pozitive ale acestei soluţii, pe care le presupunem, fără a restrînge generalitatea, a fi primele s, $(s \leqslant n)$. Pentru că programul (P) admite o soluţie, de cîte ori vom aplica un procedeu de explicitare vom obţine o formă explicită, de tipul (3.1) — (3.3).

Scanned with OKEN Scanner

Să presupunem că se aplică programului (P) un procedeu de explicitare, prin care se urmărește explicitarea mai întîi în raport cu primele s variabile și apoi, dacă s-a reușit explicitarea în raport cu $r \leqslant m$ dintre ele, $(r \leqslant s)$, iar $r < m$, se urmărește explicitarea în raport cu $m - r$ alte variabile. Să presupunem că cele r variabile principale dintre primele s sînt primele r, ceea ce putem face fără a restrînge generalitatea; fie (3.1) —(3.3) forma explicită obținută cu $\{1, 2, \ldots, r\} \subset I$. Deosebim două cazuri: $s = r$ și $s > r$. În primul caz este evident că $\hat{x}$ este soluția de bază asociată formei explicite obținute și teorema este adevărată. Rămîne să ne ocupăm în continuare de cazul al doilea și pentru aceasta facem cîteva observații prealabile.

Dacă $r < m$, cum $r < s$ în cazul al doilea, adică nu se poate face explicitarea în raport cu nici una dintre variabilele $x_{r+1}, \ldots, x_s$, rezultă că ultimele $m - r$ ecuații nu conțin primele s variabile. În consecință, dacă într-o soluție a programului liniar vom modifica numai valorile primelor s variabile, cu scopul de a se obține o nouă soluție, ultimele $m - r$ ecuații vor continua să fie satisfăcute și va trebui doar ca după modificare să rămînă satisfăcute primele r ecuații

$$x_i + \sum_{k \in K} \alpha_i^k x_k = \beta_i, \quad (i = 1, \ldots, r). \tag{3.4}$$

Vectorul $\bar{x}(\theta)$ construit din soluția optimă $\hat{x}$ prin

$$\bar{x}_j = \begin{cases} \hat{x}_j - \theta\, \alpha_j^{r+1} & \text{cînd } j = 1, \ldots, r, \\ \hat{x}_{r+1} + \theta & \text{cînd } j = r+1, \\ \hat{x}_j & \text{cînd } j = r+2, \ldots, n, \quad (j = 1, \ldots, n), \end{cases} \tag{3.5}$$

este soluție a programului liniar (P) pentru orice valoare a lui θ, deoarece

$$\bar{x}_i + \sum_{k \in K} \alpha_i^k \bar{x}_k = \hat{x}_i + \sum_{k \in K} \alpha_i^k \hat{x}_k, \quad (i = 1, \ldots, r). \tag{3.6}$$

Scanned with OKEN Scanner

Dacă luăm θ suficient de mic în valoare absolută, fie pozitiv fie negativ, vectorul $\bar{x}(\theta)$ va fi chiar soluţie admisibilă a programului liniar. Într-adevăr, dacă definim

$$R^+ = \{i \mid i \in \{1, \ldots, r\},\ \alpha_i^{r+1} > 0\},$$
$$R^- = \{i \mid i \in \{1, \ldots, r\},\ \alpha_i^{r+1} < 0\}, \tag{3.7}$$

şi

$$\theta_3 = \begin{cases} \max\limits_{i \in R^-} \dfrac{\hat{x}_i}{\alpha_i^{r+1}} & \text{cînd } R^- \neq \emptyset, \\ -\infty & \text{cînd } R^- = \emptyset, \end{cases}$$

$$\theta_1 = \max\ (-\hat{x}_{r+1}, \theta_0), \tag{3.8}$$

$$\theta_2 = \begin{cases} \min\limits_{i \in R^+} \dfrac{\hat{x}_i}{\alpha_i^{r+1}} & \text{cînd } R^+ \neq \emptyset, \\ +\infty & \text{cînd } R^+ = \emptyset, \end{cases}$$

condiţiile de admisibilitate ale soluţiei $\bar{x}(\theta)$ se scriu $\theta \in [\theta_1, \theta_2]$ cînd $R^+ \neq \emptyset$ şi $\theta \in [\theta_1, +\infty)$ cînd $R^+ = \emptyset$, şi din definiţiile (3.8) este clar că în ambele cazuri intervalul de admisibilitate conţine un interval suficient de mic cu centrul în origine. Să mai observăm că valoarea corespunzătoare a funcţiei obiectiv este

$$\bar{f}(\theta) = \hat{f} + \theta \alpha_0^{r+1}; \tag{3.9}$$

de aceea α_0^{r+1} trebuie să fie nul, altfel luînd θ suficient de mic şi de semn contrar lui α_0^{r+1}, am obţine o soluţie admisibilă $\bar{x}(\theta)$ pentru care $\bar{f}(\theta) < \hat{f}$, ceea ce ar contrazice optimalitatea soluţiei x. Avînd în vedere acest ultim fapt, rezultă că vectorul $\bar{x}(\theta)$ definit prin (3.5) este o soluţie optimă a programului liniar pentru orice θ din domeniul de admisibilitate definit mai sus. În particular, dacă $\theta = \theta_1$, am construit o soluţie optimă nouă; această soluţie

are proprietate că cel mult $s - 1$ componente ale sale sînt pozitive. Acum, fie că noua soluţie optimă este o soluţie de bază, fie că putem relua aplicarea procedeului de construcţie a unei noi soluţii optime. Cum vectorul nul, dacă este soluţie optimă, este şi soluţie optimă de bază, rezultă că după un număr finit de paşi vom construi pe această cale o soluţie optimă de bază. Teorema este demonstrată.

Din teorema 2.1 se ştie că numărul soluţiilor de bază ale unui program liniar ce admite o formă explicită este finit. Prin ipoteza teoremei 3.1 restricţiile programului liniar sînt compatibile, deci programul liniar admite o formă explicită şi, prin urmare, el are un număr finit de soluţii de bază; teorema 3.1 precizează că, în ipoteza existenţii unei soluţii optime, cel puţin una dintre soluţiile de bază este optimă. Atunci, dacă dorim să găsim o soluţie optimă a programului liniar este suficient s-o căutăm în mulţimea finită a soluţiilor de bază, în loc de a o căuta în mulţimea infinită a soluţiilor admisibile ale programului. Această concluzie importantă se află la baza metodelor generale de rezolvare pe care le vom expune mai departe.

Se impun totuşi două observaţii. Atunci cînd se propune spre rezolvare un program liniar, nu se ştie dacă există o soluţie optimă, dar, în general, există foarte multe soluţii de bază admisibile şi ar fi uneori o eroare să luăm drept soluţie optimă pe cea mai bună dintre soluţiile de bază admisibile. Această situaţie este ilustrată prin exemplul 3.1. Chiar dacă o soluţie optimă există şi putem să ne limităm căutarea la cercetarea mulţimii finite a soluţiilor de bază, problema este încă departe de a fi rezolvată, întrucît lista soluţiilor de bază poate fi foarte lungă şi astfel destul de greu de realizat numeric.

Exemplul 3.1. Considerăm programul liniar: să se minimizeze

$$f = x - 2y, \tag{3.10}$$

cu restricţiile

$$\left.\begin{aligned} x - y + u &= 1 \\ -x + y + v &= 1 \end{aligned}\right\} \tag{3.11}$$

şi

$$x \geqslant 0, \quad y \geqslant 0, \quad u \geqslant 0, \quad v \geqslant 0. \tag{3.12}$$

Lista soluţiilor de bază şi a valorilor corespunzătoare ale funcţiei obiectiv este dată în tabelul următor:

	x	y	u	v	f
S_1	0	0	1	1	0
S_2	−1	0	2	0	−1
S_3	0	1	2	0	−2
S_4	1	0	0	2	1
S_5	0	−1	0	2	2

Ar părea că soluţia de bază S_3 este soluţia optimă a programului liniar şi aşa ar fi dacă programul ar avea soluţie optimă. Dar se verifică imediat că formulele

$$x = \theta, \quad y = 1 + \theta, \quad u = 2, \quad v = 0 \tag{3.13}$$

dau o familie de soluţii admisibile cînd $\theta \in [0, +\infty)$. Valoarea funcţiei obiectiv corespunzătoare unei soluţii admisibile din familie este $f = -2 - \theta$ şi, prin urmare, funcţia obiectiv poate lua valori oricît de mici pe mulţimea de soluţii admisibile (3.13). Este clar că nu există o soluţie admisibilă pentru care funcţia obiectiv să ia o valoare minimă, deci nu există soluţie optimă şi nu putem aplica teorema 3.1, cu toate că putem găsi „cea mai bună" soluţie de bază admisibilă, anume S_3.

Metodele generale de rezolvare a programelor liniare examinează lista soluţiilor de bază, pun în evidenţă cazurile cînd nu există soluţii optime, iar atunci cînd există soluţii optime determină o soluţie de bază optimă prin examinarea numai parţială a listei soluţiilor de bază. Pentru justificarea algoritmilor cuprinşi în aceste metode mai sînt încă necesare rezultatele de mai jos.

Teorema 3.2. *Dacă* (P) *este un program liniar sub o formă explicită primal admisibilă,* (3.1) — (3.3) *cu* $\beta_i \geqslant 0$, $(i \in I)$, *şi pentru un anumit indice* $h \in K$ *avem* $\alpha_0^h < 0$ *şi pentru toţi* $i \in I$ *avem* $\alpha_i^h \leqslant 0$, *atunci programul liniar* (P) *are soluţii admisibile ce dau valori oricît de mici ale funcţiei obiectiv.*

Demonstraţie. Din (3.2) să remarcăm că vectorul $x(\theta)$ dat prin

$$\begin{aligned} x_i &= \beta_i - \alpha_i^h\, \theta, \quad (i \in I), \\ x_h &= \theta, \; x_k = 0, \quad (k \in K, \; k \neq h), \end{aligned} \tag{3.14}$$

reprezintă în condiţiile teoremei o familie de soluţii admisibile cînd $\theta \in [0, +\infty)$. Valorile corespunzătoare ale funcţiei obiectiv sînt

$$f(\theta) = f_0 + \alpha_0^h \theta. \tag{3.15}$$

Cum $\alpha_0^h < 0$, dat fiind un număr oricît de mic α, orice soluţie admisibilă din familia $x(\bar{\theta})$ cu

$$\bar{\theta} > \frac{\alpha - f_0}{\alpha_0^h} \tag{3.16}$$

va avea valoarea corespunzătoare a funcţiei obiectiv $f(\bar{\theta}) < \alpha$.

Să presupunem că dorim să realizăm un pivotaj primal asupra unui program liniar aflat sub o formă explicită primal admisibilă, cu scopul de a determina o soluţie de bază mai bună decît cea corespunzătoare formei explicite disponibile şi nu putem face acest pivotaj primal, fiindcă întîlnim situaţia (II). Din teorema 3.2 rezultă că efortul de a obţine o asemenea soluţie de bază în vederea rezolvării programului liniar este inutil, deoarece programul liniar considerat nu are soluţie optimă.

Teorema 3.3. *Dacă* (P) *este un program liniar sub o formă explicită primal admisibilă,* (3.1) — (3.3) *cu* $\beta_i \geqslant 0$, $(i \in I)$, *şi pentru toţi* $k \in K$ *avem* $\alpha_0^k \geqslant 0$, *atunci soluţia de bază corespunzătoare formei explicite este o soluţie optimă a lui* (P).

Demonstraţie. Să remarcăm că valoarea funcţiei obiectiv (3.1) pentru orice soluţie admisibilă a programului liniar este cel puţin egală cu f_0, căci toţi coeficienţii α_0^k sînt nenegativi. Cum valoarea funcţiei obiectiv pentru soluţia de bază admisibilă

$$x_i = \beta_i, \quad x_k = 0, \quad (i \in I, \quad k \in K), \tag{3.17}$$

corespunzătoare formei explicite disponibile, este egală cu f_0, deducem că (3.17) este o soluţie optimă a programului liniar (P).

Să presupunem din nou că dorim să realizăm un pivotaj primal, asupra unui program liniar aflat sub o formă explicită primal admisibilă, cu scopul de a determina o soluţie de bază „mai bună" decît cea corespunzătoare formei explicite disponibile şi nu putem face acest pivotaj primal, fiindcă întîlnim situaţia (I). Din teorema 3.3 rezultă că efortul de a obţine o asemenea soluţie de bază este inutil, deoarece soluţia de bază disponibilă este o soluţie optimă.

Scanned with OKEN Scanner

Aşadar, în baza teoremelor 3.2, 3.3 şi a observaţiei finale din § 2.5 rezultă teorema următoare :

Teorema 3.4. *Dacă programul liniar (P) aflat sub o formă explicită primal admisibilă nu are soluţii de bază degenerate, atunci după un număr finit de pivotaje primale fie se va găsi o soluţie optimă, fie se va stabili inexistenţa unei asemenea soluţii.*

Teoremele deja prezentate permit formularea unui algoritm pentru rezolvarea unui program liniar aflat sub o formă explicită primal admisibilă, prin pivotaje primale. Înainte de a trece la formularea algoritmului să dăm şi rezultate analoage relative la programe liniare aflate sub o formă explicită dual admisibilă.

Teorema 3.5. *Dacă (P) este un program liniar sub o formă explicită dual admisibilă, (3.1) — (3.3) cu* $\alpha_0^k \geqslant 0$, $(k \in K)$, *şi pentru un anumit* $j \in I$ *avem* $\beta_j < 0$ *şi pentru toţi* $k \in K$ *avem* $\alpha_j^k \geqslant 0$, *atunci programul liniar (P) nu are soluţii admisibile.*

Demonstraţie. Să considerăm ecuaţia

$$x_j + \sum_{k \in K} \alpha_j^k x_k = \beta_j, \tag{3.18}$$

din sistemul restricţiilor; deoarece membrul drept este negativ şi toţi coeficienţii variabilelor din membrul stîng sînt nenegativi, nici un sistem de valori nenegative ale variabilelor nu poate satisface ecuaţia (3.18).

Să presupunem că dorim să realizăm un pivotaj dual asupra unui program liniar aflat sub o formă explicită dual admisibilă şi nu putem face acest pivotaj, fiindcă întîlnim situaţia (II′); din teorema 3.5 rezultă că în schimb am stabilit inexistenţa soluţiilor admisibile şi problema este rezolvată.

Teorema 3.6. *Dacă (P) este un program liniar sub o formă explicită dual admisibilă, (3.1) — (3.3) cu* $\alpha_0^k \geqslant 0$, $(k \in K)$, *şi pentru toţi* $i \in I$ *avem* $\beta_i \geqslant 0$, *atunci soluţia de bază corespunzătoare formei explicite este o soluţie optimă a lui (P).*

Demonstraţie. Concluzia rezultă din teorema 3.3, ale cărei ipoteze coincid cu cele ale teoremei 3.6.

Să presupunem că dorim să realizăm un pivotaj dual asupra unui program liniar aflat sub o formă explicită dual admisibilă şi nu putem face acest pivotaj, fiindcă întîlnim situaţia (I′); din teorema 3.6 rezultă că în schimb am stabilit optimalitatea soluţiei de bază corespunzătoare formei explicite disponibile şi problema este rezolvată.

Scanned with OKEN Scanner

Așadar, în baza teoremelor 3.5, 3.6 și a observației finale din § 2.6 rezultă teorema următoare :

Teorema 3.7. *Dacă programul liniar* (P) *aflat sub o formă explicită dual admisibilă nu are soluții de bază degenerate dual, atunci după un număr finit de pivotaje duale fie vom găsi o soluție optimă, fie vom stabili inexistența soluțiilor admisibile.*

Cu aceasta dispunem și de toate rezultatele necesare pentru formularea unui algoritm de rezolvare a unui program liniar, aflat sub o formă explicită dual admisibilă, prin pivotaje duale.

3.2. ALGORITMUL SIMPLEX PRIMAL PENTRU REZOLVAREA PROGRAMELOR LINIARE AFLATE SUB O FORMĂ EXPLICITĂ PRIMAL ADMISIBILĂ

Fie (P) un program liniar aflat sub o formă explicită primal admisibilă (F^0), iar (S^0) soluția de bază admisibilă corespunzătoare formei explicite. Așa cum arătam în paragraful precedent, prin algoritmul simplex primal se urmărește examinarea numai parțială a listei soluțiilor de bază ale sistemului de restricții, în vederea găsirii unei soluții de bază optime sau stabilirii inexistenței soluțiilor optime.

Ideile de bază ale algoritmului simplex primal sînt : a) se examinează numai soluții de bază admisibile ; b) se parcurge sublista soluțiilor de bază admisibile trecîndu-se în fiecare pas, dacă este posibil, de la o soluție de bază admisibilă la o altă soluție de bază admisibilă pentru care valoarea funcției obiectiv este mai mică decît pentru prima. Algoritmul se realizează prin aplicarea în fiecare pas a unui pivotaj primal cu pivotul determinat în cursul pasului. Justificarea algoritmului este dată prin teoremele 3.1 — 3.4. Se presupune că programul liniar (P) nu are soluții de bază degenerate, pentru a fi asigurată convergența. Rămîne să formulăm acum operațiile unui pas al algoritmului.

Să admitem că după r pași programul liniar (P) se găsește sub o formă explicită primal admisibilă (F^r), $(r \geqslant 0)$, și pentru a nu mai complica notațiile fie (3.1) — (3.3) această formă, unde $I = I^r$ și $K = K^r$ sînt mulțimile de indici ale variabilelor principale și respectiv secundare, iar $f_0 = f^r$ este valoarea funcției obiectiv pentru soluția de bază (S^r) corespunzătoare formei explicite. Atunci operațiile pasului $r + 1$ sînt următoarele :

1° aplicarea criteriului de optimalitate (teorema 3.3) : se stabilește dacă $\alpha_0^k \geqslant 0$ pentru toți $k \in K^r$; în caz afirmativ soluția de

bază (S^r) este optimă și algoritmul este oprit, în caz contrar se trece mai departe;

2° aplicarea criteriului de intrare [prima regulă (2.50)]: se determină variabila secundară x_h ce devine principală, cu ajutorul regulii

$$\alpha_0^h = \min_k \alpha_0^k, \tag{3.19}$$

și se stabilește dacă $\alpha_i^h \leqslant 0$ pentru toți $i \in I^r$; în caz afirmativ programul liniar nu are soluție optimă (teorema 3.2) și algoritmul este oprit, iar în caz contrar se trece mai departe;

3° aplicarea criteriului de ieșire [a doua regulă (2.50)]: se determină variabila principală x_j ce devine secundară, cu ajutorul regulii

$$\frac{\beta_j}{\alpha_j^h} = \min_{i \mid \alpha_i^h > 0} \frac{\beta_i}{\alpha_i^h}, \tag{3.20}$$

și se trece mai departe;

4° realizarea unui pivotaj cu pivotul α_j^h.

Se înțelege, programarea algoritmului pentru calculator conduce la întrepătrunderea operațiilor 1 — 4 precum și la introducerea unor operații suplimentare impuse de tehnica de realizare a programului. De exemplu, operația 1 se realizează determinîndu-se numărul α_0^h din (3.19) și stabilindu-se dacă $\alpha_0^h \geqslant 0$ sau $\alpha_0^h < 0$; indicele h este determinat odată cu α_0^h; apoi cercetarea numerelor α_i^h cu $i \in I^r$ se face paralel cu aplicarea criteriului de ieșire etc. Este necesar de asemenea a se verifica dacă nu cumva pivotul α_j^h determinat este prea mic încît prin împărțire la el să se obțină numere a căror mărime împiedică realizarea în continuare a programului în calculator. De multe ori, dimensiunea mare a problemei impune folosirea unor programe mai elaborate care fac apel la datele strict necesare pentru rezolvarea problemei.

Exemplul 3.2. Considerăm programul liniar: să se minimizeze

$$f = 5 - 2x_1 - 3x_3, \tag{3.21}$$

cu restricțiile

$$\left.\begin{aligned} -x_1 + x_2 + x_3 &= 1 \\ 2x_1 - x_3 + x_4 &= 2 \\ -x_1 + x_3 + x_5 &= 3 \end{aligned}\right\} \tag{3.22}$$

și

$$x_i \geqslant 0, \quad (i = 1, \ldots, 5). \tag{3.23}$$

Programul liniar considerat este sub o formă explicită primal admisibilă; soluția de bază corespunzătoare formei explicite este $(S^0): x_1 = 0,\ x_2 = 1,\ x_3 = 0,\ x_4 = 2,\ x_5 = 3$, iar valoarea funcției obiectiv pentru această soluție este $f^0 = 5$. Rezolvarea programului liniar se face prin algoritmul simplex primal. Realizăm pașii algoritmului scriind în locul matricelor lărgite ale formelor explicite succesive niște tabele care conțin exact aceste matrice, dar în partea stîngă indică variabilele principale corespunzătoare liniilor, iar în partea superioară variabilele corespunzătoare coloanelor pentru fiecare matrice.

Matricea lărgită $\bar{\mathscr{A}}^0$ a programului liniar este conținută în tabelul

	x_1	x_2	x_3	x_4	x_5	
f	−2	0	−3	0	0	−5
x_2	−1	1	①	0	0	1
x_4	2	0	−1	1	0	2
x_5	−1	0	1	0	1	3

Pasul 1. În prima linie sînt elemente negative, deci putem continua aplicarea algoritmului; elementul minim fiind −3, variabila x_3 devine principală, iar raportul minim dintre 1/1 și 3/1 fiind primul, variabila principală din ecuația ai cărei coeficienți dau acest raport, adică x_2, devine secundară. Realizăm un pivotaj primal cu pivotul $\pi^1 = \alpha_2^3 = 1$ și obținem

	x_1	x_2	x_3	x_4	x_5	
f	−5	3	0	0	0	−2
x_3	−1	1	1	0	0	1
x_4	①	1	0	1	0	3
x_5	0	−1	0	0	1	2

tabel care conține matricea lărgită $\bar{\mathscr{A}}^1$. Noua soluție de bază admisibilă este $(S^1): x_1 = 0,\ x_2 = 0,\ x_3 = 1,\ x_4 = 3,\ x_5 = 2$, cu valoarea corespunzătoare a funcției obiectiv $f^1 = 2$.

Pasul 2. În prima linie este un element negativ, deci putem continua aplicarea algoritmului; unicul element negativ este coeficient al variabilei x_1 care devine principală, iar raportul 3/1 fiind unicul raport cu numitor pozitiv, variabila principală

din ecuația ai cărei coeficienți dau acest raport, adică x_4, devine secundară. Realizăm un pivotaj primal cu pivotul $\pi^2 = 1$ și obținem

	x_1	x_2	x_3	x_4	x_5	
f	0	8	0	5	0	13
x_3	0	2	1	1	0	4
x_1	1	1	0	1	0	3
x_5	0	−1	0	0	1	2

tabel care conține matricea lărgită $\overline{\mathscr{A}}^2$. Noua soluție de bază admisibilă este (S^2) : $x_1 = 3$, $x_2 = 0$, $x_3 = 4$, $x_4 = 0$, $x_5 = 2$, cu valoarea corespunzătoare a funcției obiectiv $f^2 = -13$.

Pasul 3. (S^2) este o soluție de bază optimă, căci toate elementele primei linii sînt nenegative, programul liniar este rezolvat.

Remarcăm că exemplul 2.4 a conținut de asemenea rezolvarea unui program liniar cu ajutorul metodei simplex primale.

3.3. ALGORITMUL SIMPLEX DUAL PENTRU REZOLVAREA PROGRAMELOR LINIARE AFLATE SUB O FORMĂ EXPLICITĂ DUAL ADMISIBILĂ

Fie (P) un program liniar aflat sub o formă explicită dual admisibilă (F^0), iar (S^0) soluția de bază dual admisibilă corespunzătoare formei explicite. Ca și prin algoritmul simplex primal, prin algoritmul simplex dual se urmărește examinarea numai parțială a listei soluțiilor de bază ale sistemului de restricții, de această dată în vederea găsirii unei soluții de bază optime sau stabilirii inexistenței soluțiilor admisibile.

Ideile de bază ale algoritmului simplex dual sînt : a′) se examinează numai soluții de bază dual admisibile ; b′) se parcurge sublista soluțiilor de bază dual admisibile, trecîndu-se în fiecare pas, dacă este posibil, de la o soluție de bază dual admisibilă la o altă soluție de bază dual admisibilă pentru care valoarea funcției obiectiv este mai mare decît pentru prima. Algoritmul se realizează prin aplicarea în fiecare pas a unui pivotaj dual cu pivotul determinat în cursul pasului. Justificarea algoritmului este dată prin teoremele 3.1, 3.5 — 3.7. Se presupune că programul liniar (P) nu are soluții de bază degenerate dual, pentru a fi asigurată convergența. Rămîne să formulăm acum operațiile unui pas al algoritmului.

Să admitem că după r pași, programul liniar (P) se găsește sub o formă explicită dual admisibilă (F^r), $(r \geqslant 0)$, și pentru a nu

mai complica notaţiile, fie (3.1) — (3.3) această formă, unde $I = I^r$ şi $K = K^r$ sînt mulţimile de indici ale variabilelor principale şi respectiv secundare, iar $f_0 = f^r$ este valoarea funcţiei obiectiv pentru soluţia de bază (S^r) corespunzătoare formei explicite. Atunci operaţiile pasului $r+1$ sînt următoarele:

1° aplicarea criteriului de optimalitate (teorema 3.6): se stabileşte dacă $\beta_i \geqslant 0$ pentru toţi $i \in I^r$; în caz afirmativ soluţia de bază (S^r) este optimă şi algoritmul este oprit, în caz contrar se trece mai departe;

2° aplicarea criteriului de ieşire [prima regulă (2.62)]: se determină variabila principală x_j ce devine secundară, cu ajutorul regulii

$$\beta_j = \min_i \beta_i, \tag{3.24}$$

şi se stabileşte dacă $\alpha_j^k \geqslant 0$ pentru toţi $k \in K$; în caz afirmativ programul liniar nu are soluţii admisibile (teorema 3.5) şi algoritmul este oprit, în caz contrar se trece mai departe;

3° aplicarea criteriului de intrare [a doua regulă (2.62)]: se determină variabila secundară x_h ce devine principală cu ajutorul regulii

$$\frac{\alpha_0^h}{\alpha_j^h} = \min_{k \mid \alpha^k < 0} \left| \frac{\alpha_0^k}{\alpha_j^k} \right|, \tag{3.25}$$

şi se trece mai departe;

4° realizarea unui pivotaj cu pivotul α_j^h.

Se înţelege, asupra intrepătrunderii operaţiilor 1—4 într-un program pentru calculator şi asupra necesităţilor de introducere a unor operaţii suplimentare într-un asemenea program se pot face observaţii asemănătoare celor relative la algoritmul simplex primal.

Exemplul 3.3. Considerăm programul liniar: să se minimizeze

$$f = -5 + x_2 + 2x_4 + 3x_5, \tag{3.26}$$

cu restricţiile

$$\left.\begin{aligned} x_1 + x_2 \quad - 2x_4 + x_5 &= -2 \\ -x_2 + x_3 + x_4 - x_5 &= -3 \end{aligned}\right\} \tag{3.27}$$

și

$$x_i \geqslant 0, \quad (i = 1, \ldots, 5). \tag{3.28}$$

Programul liniar considerat este sub o formă explicită dual admisibilă; soluția de bază corespunzătoare formei explicite este (S^0): $x_1 = -2$, $x_2 = 0$, $x_3 = -3$, $x_4 = 0$, $x_5 = 0$, iar valoarea funcției obiectiv pentru această soluție este $f = -5$. Rezolvarea programului liniar o realizăm prin algoritmul simplex dual, folosind ca și în exemplul precedent tabele ce conțin matricele lărgite ale formelor explicite succesive.

Matricea lărgită $\bar{\mathcal{A}}^0$ a programului liniar este conținută în tabelul următor:

	x_1	x_2	x_3	x_4	x_5	
f	0	1	0	2	3	5
x_1	1	1	0	−2	1	−2
x_3	0	(−1)	1	1	−1	−3

Pasul 1. În ultima coloană sînt elemente negative, deci putem continua aplicarea algoritmului; elementul minim fiind -3, variabila x_3 devine secundară, iar raportul minim în valoare absolută dintre $1/(-1)$ și $3/(-1)$ fiind primul, variabila secundară ai cărei coeficienți dă acest raport, adică x_2, devine principală. Realizăm un pivotaj dual cu pivotul $\pi^1 = \alpha_3^2 = -1$ și obținem tabelul

	x_1	x_2	x_3	x_4	x_5	
f	0	0	1	3	2	2
x_1	1	0	1	(−1)	0	−5
x_2	0	1	−1	−1	1	3

care conține matricea lărgită $\bar{\mathcal{A}}^1$. Noua soluție de bază dual admisibilă este (S^1): $x_1 = -5$, $x_2 = 3$, $x_3 = 0$, $x_4 = 0$, $x_5 = 0$, cu valoarea corespunzătoare a funcției obiectiv $f^1 = -2$.

Pasul 2. În ultima coloană este un element negativ, deci putem continua aplicarea algoritmului; unicul element negativ -5 este termen liber în ecuația unde x_1 este variabilă principală, astfel că aceasta va deveni secundară; unicul coeficient negativ din ecuație este α_1^4, deci variabila secundară x_4 devine principală. Realizăm un pivotaj dual cu pivotul $\pi^2 = -1$ și obținem

	x_1	x_2	x_3	x_4	x_5	
f	3	0	4	0	2	−13
x_4	−1	0	−1	1	0	5
x_2	−1	1	−2	0	1	8

tabel care conține matricea lărgită $\overline{\mathcal{B}}^2$. Noua soluție de bază dual admisibilă este (S^2): $x_1 = 0$, $x_2 = 8$, $x_3 = 0$, $x_4 = 5$, $x_5 = 0$, cu valoarea corespunzătoare a funcției obiectiv $f^2 = 13$.

Pasul 3. (S^2) este o soluție de bază optimă, căci toate elementele ultimei coloane sînt nenegative, programul liniar este rezolvat.

Remarcăm că exemplul 2.5 a conținut de asemenea rezolvarea unui program liniar cu ajutorul algoritmului simplex dual.

Să clarificăm un aspect al ideii de bază a algoritmului simplex dual. Cînd comparăm două soluții de bază ale unui program liniar, sîntem tentați a declara că prima este „mai bună" decît a doua, dacă valoarea funcției obiectiv pentru prima soluție este mai mică decît pentru a doua. Această accepțiune a expresiei „mai bună" este pe deplin justificată în cazul cînd se compară două soluții de bază admisibile. Dacă este însă vorba de a se compara două soluții de bază dual admisibile, am face o eroare admițînd aceeași accepțiune a expresiei „mai bună". Într-adevăr, să considerăm două soluții de bază dual admisibile cu valorile funcției obiectiv f' și f'', $f' < f''$, și să ținem seamă că, în bază admisibilității duale, coeficienții funcției obiectiv fiind nenegativi, orice soluție admisibilă, deci și una optimă, are valoarea funcției obiectiv cel puțin egală cu f''; este clar că soluția de bază pentru care valoarea funcției obiectiv este f' este mai depărtată de o soluție optimă decît cea pentru care valoarea funcției obiectiv este f'', căci $f' < f''$. De aceea, trebuie să adoptăm pentru expresia „mai bună" accepțiunea „mai apropiată de o soluție optimă". Cu această accepțiune este clar că atît algoritmul simplex primal cît și algoritmul simplex dual construiesc în fiece pas nefinal o soluție de bază „mai bună" decît cea cunoscută la începutul pasului.

3.4. REZOLVAREA UNUI PROGRAM LINIAR SUB FORMA STANDARD, PRIN METODA SIMPLEX PRIMALĂ SAU DUALĂ

Considerăm un program liniar (P) cu forma standard: să se minimizeze

$$f = \sum_{k=1}^{n} c_k x_k, \tag{3.29}$$

cu restricțiile

$$\sum_{k=1}^{n} a_i^k x_k = b_i, \quad (i = 1, \ldots, m), \tag{3.30}$$

și

$$x_k \geqslant 0, \quad (k = 1, \ldots, n). \tag{3.31}$$

Am văzut în § 2.2 că orice problemă de maximizare se poate transforma într-o problemă de minimizare și că orice inecuație din sistemul restricțiilor poate fi înlocuită printr-o ecuație, introducînd o variabilă suplimentară nenegativă; deci, un program liniar oarecare se poate aduce la forma (3.29) — (3.31).

În general, problema (P) nu se găsește sub o formă explicită, sau chiar dacă (P) este sub o formă explicită, ea nu este numaidecît primal sau dual admisibilă. De aceea, rezolvarea programului (P) trebuie făcută în general în două faze cu sarcini deosebite: în prima fază se urmărește aducerea programului la o formă explicită primal sau dual admisibilă, iar în faza a doua se urmărește rezolvarea programului liniar explicitat. Pînă acum ne-am ocupat de fapt de faza a doua a rezolvării unui program liniar; în continuare, vom trece la prezentarea fazei întîi. Un fapt important de subliniat: deși scopul primei faze nu este de a rezolva programul liniar dat, totuși pentru găsirea unei forme explicite primal (dual) admisibile se întrebuințează tot algoritmul simplex primal (dual); aceasta se întîmplă fiindcă explicitarea programului liniar dat se realizează în faza întîi prin rezolvarea unui program liniar ajutător.

Se consideră de obicei că ansamblul tuturor operațiilor necesare pentru a rezolva un program liniar sub forma standard, cu ajutorul algoritmului simplex primal (dual), în două faze în care algoritmul se aplică unor probleme diferite, constituite metoda simplex primală (duală). În această accepțiune metoda simplex primală sau duală sînt concepute ca algoritmi complecși, a căror subrutină principală este algoritmul simplex primal sau dual, respectiv.

3.4.1. *OBȚINEREA UNEI FORME EXPLICITE PRIMAL ADMISIBILE CU AJUTORUL ALGORITMULUI SIMPLEX PRIMAL*

Programului liniar (P) dat sub forma (3.29) — (3.31) i se asociază un program liniar ajutător (P') de forma: să se minimizeze

$$g = \sum_{i=1}^{m} y_i, \tag{3.32}$$

cu restricţiile

$$\sum_{k=1}^{n} a_i^k x_k + y_i = b_i, \quad (i = 1, \ldots, m), \tag{3.33}$$

şi

$$y_i \geqslant 0, \; x_k \geqslant 0, \; (i = 1, \ldots, m; \; k = 1, \ldots, n). \tag{3.34}$$

Variabilele introduse y_i, $(i = 1, \ldots, m)$, sînt denumite *variabile artificiale*. Putem presupune $b_i \geqslant 0$ pentru toţi $i = 1, \ldots, m$ fără a restrînge generalitatea, deoarece dacă vreun termen liber ar fi fost negativ într-o ecuaţie (3.30) am fi înmulţit acea ecuaţie cu -1 înainte de introducerea variabilei artificiale corespunzătoare. Din sistemul de restricţii (3.33) deducem că pentru orice soluţie a problemei avem

$$\sum_{i=1}^{m} y_i = \sum_{i=1}^{m} b_i - \sum_{i=1}^{m} \sum_{k=1}^{n} a_i^k \, x_k, \tag{3.35}$$

şi, prin urmare, funcţia liniară g poate fi scrisă sub forma echivalentă

$$g = \sum_{i=1}^{m} b_i - \sum_{i=1}^{m} \sum_{k=1}^{n} a_i^k \, x_k. \tag{3.36}$$

Folosindu-ne de aceste două observaţii, remarcăm următoarele proprietăţi ale problemei (P'):

1° Programul liniar (P') dat prin (3.33), (3.34), (3.36) este sub o formă explicită primal admisibilă, anume variabilele artificiale sînt variabile principale iar variabilele iniţiale sînt variabilele secundare; ca o consecinţă a acestei proprietăţi putem trece la rezolvarea programului (P') cu algoritmul simplex primal.

2° Programul liniar (P') are totdeauna soluţie optimă; într-adevăr, ţinînd seamă de proprietatea precedentă şi de faptul că $g \geqslant 0$ pe mulţimea soluţiilor admisibile ale programului, este certă obţinerea unei soluţii optime la sfîrşitul aplicării algoritmului simplex primal.

3°. Dacă programul liniar (P) are o soluţie admisibilă $(\tilde{x}_1, \ldots, \tilde{x}_n)$, atunci $(\bar{x}_1, \ldots, \bar{x}_n, 0, \ldots, 0)$ este o soluţie optimă a programului liniar (P'); într-adevăr, $(\bar{x}_1, \ldots, \bar{x}_n, 0, \ldots, 0)$ va fi o soluţie admisibilă a lui (P') şi cum ea dă pentru funcţia obiectiv

Scanned with OKEN Scanner

valoarea $\bar{g} = 0$, înseamnă că este soluție optimă a lui (P'), datorită observației care a condus la demonstrarea proprietății precedente.

Avînd în vedere proprietățile formulate ale programului (P'), faza I a rezolvării programului liniar (P) constă din următoarele:

— asigurarea nenegativității termenilor liberi;

— construirea programului liniar auxiliar (P');

— rezolvarea programului liniar (P') prin algoritmul simplex primal, terminată cu obținerea unei soluții optime.

Să admitem că s-a rezolvat programul liniar auxiliar (P') și s-a obținut o soluție optimă $(\bar{x}_1, \ldots, \bar{x}_n, \bar{y}_1, \ldots, \bar{y}_m)$ cu valoarea corespunzătoare a funcției obiectiv $\bar{g} = \sum_{i=1}^{m} \bar{y}_i$. Deosebim două cazuri: a) $\bar{g} = 0$ și b) $\bar{g} \neq 0$. În baza proprietății 3° putem conchide că în cazul b) programul liniar inițial (P) nu are soluții admisibile și, prin urmare, rezolvarea lui este terminată. Rămîne să ne ocupăm numai de cazul a).

În cazul a) egalitatea $\bar{g} = 0$ implică $\bar{y}_i = 0$ pentru toți $i = 1, \ldots, m$; deși toate variabilele artificiale sînt nule în soluția optimă găsită a programului (P'), nu este sigur că aceste variabile sînt toate secundare în forma explicită finală a lui (P'). De aceea sînt posibile două situații:

a_1) toate variabilele artificiale sînt variabile secundare în forma explicită finală a programului (P');

a_2) există cel puțin o variabilă artificială care este variabilă principală în această formă explicită.

Desigur, în ultima situație soluția optimă găsită este degenerată, avînd o variabilă principală nulă.

Să examinăm cazul a_1). Deoarece revenirea de la sistemul restricțiilor programului (P') la sistemul restricțiilor programului (P) se face punînd toate variabilele artificiale egale cu zero și cum prin aceasta dispar din matricea formei explicite finale a lui (P') coloanele corespuzătoare acestor variabile, rezultă că în situația a_1) nu va dispărea în urma operației nici o coloană a matricei unitate. Așadar, după omiterea coloanelor în chestiune va rămîne matricea unui sistem explicitat echivalent sistemului restricțiilor lui (P) și totodată se va menține proprietatea de admisibilitate primală; deci, sarcina fazei I a fost realizată și urmează a se trece la faza II. Desigur, în faza a doua trebuie să reluăm funcția obiectiv f care trebuie exprimată numai cu variabilele secundare ale formei explicite, ceea ce nu prezintă complicații principiale; totuși, pentru uni-

Scanned with OKEN Scanner

tatea calculelor, coeficienții din funcția obiectiv f a programului liniar (P) se iau și în cursul fazei întîi într-o linie suplimentară a matricei supuse calculelor, astfel ca la sfîrșitul fazei funcția f să fie deja sub forma necesară la începutul fazei a doua. Se înțelege, în faza II trebuie trecut asupra liniei cu coeficienții lui f rolul avut pînă atunci de linia cu coeficienții lui g, iar ultima trebuie suprimată. Astfel, trecerea de la faza I la faza II se face, în cazul a_1), suprimînd coloanele coeficienților variabilelor artificiale și linia coeficienților variabilelor din funcția g; după aceea urmează primul pas al fazei a doua.

Exemplul 3.4. Considerăm programul liniar: să se minimizeze

$$f = x_1 - x_2 + x_3 - x_4, \tag{3.37}$$

cu restricțiile

$$\left.\begin{aligned} 2x_1 - x_2 + 2x_3 - x_4 &= 1 \\ x_1 + x_2 + x_3 + x_4 &= 1 \end{aligned}\right\} \tag{3.38}$$

și

$$x_k \geqslant 0, \ (k = 1, 2, 3, 4). \tag{3.39}$$

Cum termenii liberi în (3.38) sînt pozitivi, trecem direct la construirea programului liniar auxiliar: să se minimizeze

$$g = y_1 + y_2, \tag{3.40}$$

cu restricțiile

$$\left.\begin{aligned} 2x_1 - x_2 + 2x_3 - x_4 + y_1 &= 1 \\ x_1 + x_2 + x_3 + x_4 + y_2 &= 1 \end{aligned}\right\} \tag{3.41}$$

și

$$y_1 \geqslant 0, \ y_2 \geqslant 0, \ x_k \geqslant 0, \ (k = 1, 2, 3, 4). \tag{3.42}$$

Prin adunarea restricțiilor (3.41) obținem pentru funcția g o exprimare conținînd numai variabilele secundare ale sistemului:

$$g = -3x_1 - 3x_3 + 2. \tag{3.43}$$

Scanned with OKEN Scanner

Scriem tabelul corespunzător formei explicite disponibile a programului liniar auxiliar cu o linie suplimentară pentru coeficienții variabilelor din funcția f:

	x_1	x_2	x_3	x_4	y_1	y_2	
g	−3	0	−3	0	0	0	−2
f	1	−1	1	−1	0	0	0
y_1	②	−1	2	−1	1	0	1
y_2	1	1	1	1	0	1	1

tabel care conține matricea de plecare $\bar{\mathcal{A}}$.

F a z a I. Efectuăm un prim pivotaj și găsim tabelul

	x_1	x_2	x_3	x_4	y_1	y_2	
g	0	−3/2	0	−3/2	3/2	0	−1/2
f	0	−1/2	0	−1/2	−1/2	0	−1/2
x_1	1	−1/2	1	−1/2	1/2	0	1/2
y_2	0	③/2	0	3/2	−1/2	1	1/2

care conține matricea $\bar{\mathcal{A}}^1$, apoi un al doilea pivotaj și obținem tabelul

	x_1	x_2	x_3	x_4	y_1	y_2	
g	0	0	0	0	1	1	0
f	0	0	0	0	−2/3	−1/3	−1/3
x_1	1	0	1	0	1/3	1/3	2/3
x_2	0	1	0	1	−1/3	2/3	1/3

care conține matricea $\bar{\mathcal{A}}^2$ a formei explicite finale a programului auxiliar. Soluția optimă corespunzătoare este $x_1 = 2/3$, $x_2 = 1/3$, $x_3 = 0$, $x_4 = 0$, $y_1 = 0$, $y_2 = 0$ și pentru ea $g = 0$. Sîntem în cazul a$_1$) și pentru a trece la faza a doua se suprimă din tabel partea haşurată, adică linia cu coeficienții funcției g și coloanele corespunzătoare variabilelor artificiale y_1 și y_2.

Faza II. Se începe cu tabelul găsit după suprimările indicate, anume

	x_1	x_2	x_3	x_4	
f	0	0	0	0	−1/3
x_1	1	0	1	0	2/3
x_2	0	1	0	1	1/3

și chiar în primul pas se constată că s-a obținut o soluție optimă a programului liniar considerat inițial; soluția este $x_1 = 2/3$, $x_2 = 1/3$, $x_3 = 0$, $x_4 = 0$, pentru care $f = 1/3$.

Să examinăm cazul a_2). Dacă în cazul a_2) am proceda ca în cazul a_1), suprimînd toate coloanele corespunzătoare variabilelor artificiale din matricea formei explicite finale a programului (P'), atunci prin aceasta s-ar suprima din matrice măcar o coloană din matricea unitate, deoarece ar trebui să suprimăm și coloanele corespunzătoare variabilelor artificiale care sînt principale. Astfel, după omiterea coloanelor în chestiune n-am obține o formă explicită a sistemului restricțiilor programului (P). De aceea, înainte de a trece de la sistemul restricțiilor lui (P') la sistemul restricțiilor lui (P) în maniera indicată în cazul a_1), trebuie să încercăm a face ca toate variabilele artificiale să devină secundare, prin pivotaje care nu mai sînt, e drept, pivotaje primale.

Considerăm o ecuație care conține o variabilă artificială principală și, prin urmare, are membrul drept nul. Dacă ecuația nu conține nici o variabilă inițială, omitem ecuația, căci oricum ea va dispărea prin revenirea la problema inițială. Dacă ecuația conține vreo variabilă inițială, luăm coeficientul unei asemenea variabile ca pivot și realizăm un pivotaj; cum membrul drept al ecuației este nul, soluția de bază disponibilă nu va fi modificată, nici valoarea funcției obiectiv. Așadar, indiferent dacă ecuația considerată conține sau nu vreo variabilă inițială, fie prin suprimarea ecuației fie printr-un pivotaj obținem o formă explicită finală a programului auxiliar (P'), iar numărul variabilelor artificiale principale a scăzut cu o unitate. Continuăm aplicarea procedeului pînă cînd toate variabilele artificiale sînt secundare. Acum va trebui să procedăm ca în cazul a_1), adică vom omite toate coloanele corespunzătoare variabilelor artificiale și linia cu coeficienții funcției g și vom

trece la faza a doua. Remarcăm că existența unei ecuații care conține numai variabile artificiale înseamnă de fapt existența unei ecuații reductibile în sistemul restricțiilor programului (P).

Exemplul 3.5. Considerăm programul liniar (P): să se minimizeze

$$f = x_1 - x_2 + x_3 \tag{3.44}$$

cu restricțiile

$$\left.\begin{aligned} x_1 + \lambda x_2 + x_3 &= 1 \\ 2x_1 + x_2 + 2x_3 &= 2 \end{aligned}\right\} \tag{3.45}$$

și

$$x_1 \geqslant 0, \quad x_2 \geqslant 0, \quad x_3 \geqslant 0, \tag{3.46}$$

unde λ este un parametru real cu valori în intervalul $[1/2, 2]$.

Construim programul liniar auxiliar (P'): să se minimizeze

$$g = y_1 + y_2 \tag{3.47}$$

cu restricțiile

$$\left.\begin{aligned} x_1 + \lambda x_2 + x_3 + y_1 &= 1 \\ 2x_1 + x_2 + 2x_3 + y_2 &= 2 \end{aligned}\right\} \tag{3.48}$$

și

$$x_1 \geqslant 0, \quad x_2 \geqslant 0, \quad x_3 \geqslant 0, \quad y_1 \geqslant 0, \quad y_2 \geqslant 0. \tag{3.49}$$

Adunînd restricțiile (3.48), găsim pentru funcția g o expresie conținînd numai variabilele secundare ale sistemului, anume

$$g = 3 - 3x_1 - (\lambda + 1)\, x_2 - 3x_3. \tag{3.50}$$

Scriem tabelul corespunzător formei explicite disponibile a programului (P') cu o linie suplimentară pentru coeficienții variabilelor din funcția f:

	x_1	x_2	x_3	y_1	y_2	
g	-3	$-(\lambda+1)$	-3	0	0	-3
f	1	-1	1	0	0	0
y_1	①	λ	1	1	0	1
y_2	2	1	2	0	1	2

tabel care conține matricea de plecare $\bar{\mathscr{E}}$.

Începem faza I printr-un pivotaj primal și găsim tabelul

	x_1	x_2	x_3	y_1	y_2	
g	0	$2\lambda - 1$	0	3	0	0
f	0	$-\lambda - 1$	0	-1	0	-1
x_1	1	λ	1	1	0	1
y_2	0	$-2\lambda + 1$	0	-2	1	0

care conține matricea $\mathscr{B}^{-1}$ a formei explicite a programului (P'), căci $2\lambda - 1 \geqslant 0$ pentru $\lambda \in [1/2, 2]$. S-a obținut soluția optimală (S^1): $x_1 = 1$, $x_2 = 0$, $x_3 = 0$, $y_1 = 0$, $y_2 = 0$, iar valoarea corespunzătoare a funcției obiectiv este $g = 0$.

Deoarece $g = 0$, sîntem în cazul a) și fiindcă y_2 este o variabilă artificială principală, se prezintă cazul a_2). Ecuația care conține y_2 ca variabilă principală este

$$(1 - 2\lambda)\, x_2 - 2y_1 + y_2 = 0. \tag{3.51}$$

Pentru $\lambda = 1/2$ ecuația trebuie suprimată și se obține o formă explicită, unde toate variabilele artificiale sînt secundare; aceasta nu este o formă explicită a programului (P'), dar vom găsi imediat o formă explicită primal admisibilă a programului (P), suprimînd coloanele coeficienților variabilelor artificiale și linia coeficienților variabilelor din funcția g:

	x_1	x_2	x_3	
f	0	$-3/2$	0	-1
x_1	1	(1/2)	1	1

În acest caz, faza I s-a terminat și trecînd la faza II obținem printr-un pivotaj primal tabelul

	x_1	x_2	x_3	
f	3	0	3	2
x_2	2	1	2	2

care conține o soluție optimă a programului (P), anume: $x_1 = 0$, $x_2 = 2$, $x_3 = 0$ și valoarea corespunzătoare a funcției obiectiv este $f = -2$.

Scanned with OKEN Scanner

Pentru $\lambda \in \left(\frac{1}{2}, 2\right]$ ecuația (3.51) conține variabila x_2 a programului inițial și vom face un pivotaj asupra matricei $\mathscr{A}^1$ cu pivotul $-2\lambda + 1$; obținem tabelul

	x_1	x_2	x_3	y_1	y_2	
g	0	0	0	1	1	0
f	0	0	0	$-\frac{3}{1-2\lambda}$	$\frac{\lambda+1}{1-2\lambda}$	-1
x_1	1	0	1	$\frac{1}{1-2\lambda}$	$\frac{\lambda}{1-2\lambda}$	1
x_2	0	1	0	$\frac{2}{1-2\lambda}$	$\frac{1}{1-2\lambda}$	0

care conține o nouă formă explicită a programului (P'), unde toate variabilele artificiale sînt variabile secundare; soluția de bază corespunzătoare este tot (S^1). Faza I s-a încheiat și vom găsi o formă explicită primal admisibilă a programului (P) suprimînd din tabel partea haşurată. Chiar în primul pas al fazei a doua se constată că programul (P) a fost rezolvat, o soluție optimă este $x_1 = 1$, $x_2 = 0$, $x_3 = 0$ și valoarea funcției obiectiv este $f = 1$. Remarcăm faptul interesant că și soluția de bază $x_1 = 0$, $x_2 = 0$, $x_3 = 1$ este optimă și nu mai există altă soluție de bază a sistemului de restricții; deci, toate soluțiile de bază ale programului (P) sînt optime pentru orice $\lambda \in \left(\frac{1}{2}, 2\right]$. Se constată ușor din tabelul final al fazei că aceeași concluzie este valabilă chiar pentru orice $\lambda \in \left(\frac{1}{2}, +\infty\right)$.

3.4.2. *OBȚINEREA UNEI FORME EXPLICITE DUAL ADMISIBILE CU AJUTORUL ALGORITMULUI SIMPLEX DUAL*

În vederea aducerii programului liniar (P) la o formă explicită dual admisibilă se aduce programul în prealabil la o formă explicită, de exemplu, pe calea arătată în § 2.3. Dacă sistemul restricțiilor se dovedește incompatibil, rezolvarea programului (P) este terminată; altfel, vom obține o formă explicită, fie (3.1) — (3.3). Dacă forma explicită obținută este deja dual admisibilă, atunci vom putea trece la rezolvarea programului (P) cu ajutorul algorit-

mului simplex dual. Rămîne să considerăm cazul cînd forma explicită obţinută nu este dual admisibilă, adică $\min_k \alpha_0^k < 0$, şi să arătăm cum se procedează în faza I a rezolvării programului (P).

Considerăm un program liniar auxiliar (P') obţinut prin adăugarea unei restricţii, denumite ecuaţie artificială :

$$x_{n+1} + \sum_{k \in K} x_k = M, \tag{3.52}$$

care conţine o nouă variabilă x_{n+1} şi un parametru M reprezentat printr-un număr real oricît de mare, şi o condiţie

$$x_{n+1} \geqslant 0. \tag{3.53}$$

În calculul manual M poate rămîne nedeterminat, dar vom ţine seamă de faptul că valoarea sa este oricît de mare. Acum, programul liniar auxiliar (P') se scrie compact sub forma : să se minimizeze

$$f = f_0 + \sum_{k \in K} \alpha_0^k x_k, \tag{3.54}$$

cu restricţiile

$$x_i + \sum_{k \in K} \alpha_i^k x_k = \beta_i, \ (i \in I + \{n+1\}), \tag{3.55}$$

unde $\alpha_{n+1}^k = 1$, $(k \in K)$, $\beta_{n+1} = M$ şi

$$x_i \geqslant 0, \ x_k \geqslant 0, \ (i \in I + \{n+1\}, \ k \in K). \tag{3.56}$$

Este clar că programul (P') este şi el sub o formă explicită, care nu este dual admisibilă. Totuşi, putem obţine o formă explicită dual admisibilă a programului (P') printr-un pivotaj, dacă alegem pivotul astfel : determinăm indicele h prin

$$\alpha_0^h = \min_k \alpha_0^k, \tag{3.57}$$

şi luăm α_{n+1}^h drept pivot. Să justificăm această afirmaţie, realizînd doar eliminarea variabilei x_h din funcţia obiectiv, cu ajutorul ecuaţiei artificiale. Avem

$$x_h = -\sum_{k \in K'} x_k - x_{n+1} + M, \ K' = K - \{h\}, \tag{3.58}$$

și, introducînd în funcția obiectiv, găsim

$$f = f_0 + \alpha_0^h M + \sum_{k \in K'} (\alpha_0^k - \alpha_0^h) x_k - \alpha_0^h x_{n+1}. \qquad (3.59)$$

Cum $\alpha_0^h < 0$ și $\alpha_0^h \leqslant \alpha_0^k$ pentru toți $k \in K'$, toți coeficienții variabilelor din funcția obiectiv (3.59) de după pivotaj sînt nenegativi.

Din acest raționament și formula (3.59) remarcăm că rezultatul s-ar menține dacă în ecuația artificială s-ar lua în locul sumei variabilelor secundare numai suma variabilelor secundare cu coeficient negativ în funcția obiectiv (3.54).

Să presupunem că am realizat pivotajul descris și am obținut o formă explicită dual admisibilă a programului liniar (P'). Acum, începem să aplicăm programului auxiliar algoritmul simplex dual și construim succesiv un șir de forme explicite dual admisibile, pînă cînd se prezintă una dintre următoarele două situații mutual exclusive:

c) variabila x_{n+1} a redevenit variabilă principală;

d) variabila x_{n+1} este variabilă secundară, dar aplicarea algoritmului simplex dual este oprită, căci programul liniar auxiliar (P') este rezolvat.

Să examinăm cazul c). Dacă x_{n+1} devine variabilă principală a formei explicite obținute în urma unui pas, rezultă că x_{n+1} nu apare decît într-o singură ecuație a formei explicite; toate celelalte ecuații precum și funcția obiectiv sînt obținute prin transformări elementare numai din sistemul restricțiilor și funcția obiectiv ale programului (P), iar ecuația care conține x_{n+1} se obține din ecuația artificială și același sistem de restricții. De aceea, suprimînd ecuația ce conține x_{n+1}, se va obține un program liniar echivalent lui (P); suprimarea ecuației nu influențează admisibilitatea dual a formei explicite, prin urmare, faza I s-a încheiat și se va trece la faza II. Se înțelege, odată cu suprimarea liniei care conține ecuația în chestiune se va suprima și coloana unitară corespunzătoare variabilei x_{n+1}. Dealtfel, s-ar putea continua aplicarea algoritmului simplex dual în faza a doua asupra formei explicite finale a primei faze, fără suprimarea vreunui element.

Exemplul 3.6. Considerăm programul liniar: să se minimizeze

$$f = 5 - 2x_2 - x_3, \qquad (3.60)$$

cu restricţiile

$$\left.\begin{aligned} x_1 + x_2 + x_3 &= 5 \\ -2x_2 - x_3 + x_4 &= -3 \end{aligned}\right\} \tag{3.61}$$

şi

$$x_k \geqslant 0, \quad (k = 1, 2, 3, 4). \tag{3.62}$$

Programul liniar este deja sub o formă explicită, dar care nu este dual admisibilă. Construim programul liniar auxiliar: să se minimizeze funcţia (3.60) cu restricţiile

$$\left.\begin{aligned} x_1 + x_2 + x_3 &= 5 \\ -2x_2 - x_3 + x_4 &= -3 \\ x_2 + x_3 + x_5 &= M \end{aligned}\right\} \tag{3.63}$$

şi

$$x_k \geqslant 0, \ (k = 1, 2, 3, 4), \ x_5 \geqslant 0. \tag{3.64}$$

Realizăm pivotajul care ne conduce la o formă explicită dual admisibilă a programului auxiliar:

	x_1	x_2	x_3	x_4	x_5	
f	0	−2	−1	0	0	−5
x_1	1	1	1	0	0	5
x_4	0	−2	−1	1	0	−3
x_5	0	(1)	1	0	1	M
f	0	0	1	0	2	$2M-5$
x_1	1	0	0	0	(−1)	$-M+5$
x_4	0	0	1	1	2	$2M-3$
x_2	0	1	1	0	1	M

Faza I. Efectuăm un pivotaj dual, ţinînd seamă că M este un număr pozitiv oricît de mare şi obţinem tabelul

	x_1	x_2	x_3	x_4	x_5	
f	2	0	1	0	0	5
x_5	−1	0	0	0	1	$M-5$
x_4	2	0	1	1	0	7
x_2	1	1	1	0	0	5

care conține forma explicită dual admisibilă finală a fazei întîi, deoarece x_5 a redevenit variabilă principală. Vom trece la faza a doua asupra tabelului

	x_1	x_2	x_3	x_4	
f	2	0	1	0	5
x_4	2	0	1	1	7
x_2	1	1	1	0	5

obținut prin suprimarea liniei corespunzătoare ecuației ce conține x_5 și a coloanei cu coeficienții lui x_5.

Faza II. Constatăm chiar în primul pas că dispunem de o soluție optimă a programului liniar (P), anume: $x_1 = 0$, $x_2 = 5$, $x_3 = 0$, $x_4 = 7$, pentru care funcția obiectiv are valoarea $f = -5$.

Să examinăm cazul d). După cum se știe, rezolvarea programului liniar auxiliar (P') se poate termina într-unul din următoarele două moduri: d_1) s-a stabilit că programul (P') nu are soluții admisibile; d_2) s-a obținut o soluție optimă. În primul caz, programul liniar inițial (P) nu are soluții admisibile, deoarece dacă $(\bar{x}_1, \ldots, \bar{x}_n)$ ar fi o soluție admisibilă a lui (P) și am pune $\bar{x}_{n+1} = M - \sum_{k \in K} \bar{x}_k$, atunci $(\bar{x}_1, \ldots, \bar{x}_{n+1})$ ar fi o soluție admisibilă a lui (P'). Rămîne să examinăm cazul al doilea, în care, ținînd seamă că x_{n+1} este variabilă a formei explicite finale, rezultă atît o soluție optimă cît și o valoare optimă dependentă de M. În această formă explicită funcția obiectiv conține variabila x_{n+1} și parametrul M numai într-o pereche de termeni de forma $\alpha_0^{n+1}(x_{n+1} - M)$, cu $\alpha_0^{n+1} \geqslant 0$ din cauza admisibilității duale. Se deosebesc atunci două subcazuri: d_2') $\alpha_0^{n+1} > 0$; d_2'') $\alpha_0^{n+1} = 0$. În primul caz, soluția optimă a programului (P') este în fapt o familie de soluții, dependentă de parametrul M, admisibile pe o semidreaptă de tipul $[M_0, +\infty)$, iar valorile corespunzătoare ale funcției obiectiv, conținînd termenul $-\alpha_0^{n+1}$, pot fi făcute oricît de mici pe intervalul precizat. Omițînd din familia de soluții componenta lui x_{n+1}, se obține o familie de soluții admisibile pe același interval pentru programul (P), căci sistemul restricțiilor lui (P) este subsistem al sistemului restricțiilor lui (P'). Cum funcțiile obiectiv ale celor două programe coincid, programul liniar (P) are soluții admisibile pentru care valoarea funcției obiectiv este oricît de mică. În cazul al doilea, deși soluția optimă a programului (P') depinde de parametrul M și este, prin urmare, tot o familie de soluții admisibile, definită

pe un interval de tipul $[M_0, +\infty)$, valoarea funcției obiectiv este aceeași pentru toate soluțiile familiei, căci $\alpha_0^{n+1} = 0$ și M apare în expresia funcției obiectiv. Omițînd din familia de soluții componenta x_{n+1}, vom obține o familie de soluții optime ale programului (P) definită pe același interval și în familie găsim o soluție de bază optimă, care corespunde valorii $M = M_0$.

Exemplul 3.7. Considerăm programul liniar (P): să se minimizeze

$$f = -2x_2 - 3, \tag{3.65}$$

cu restricțiile

$$x_1 - x_2 = -1,\ x_1 \geqslant 0,\ x_2 \geqslant 0. \tag{3.66}$$

Programul liniar are deja o formă explicită dar aceasta nu este nici primal nici dual admisibilă. Să folosim algoritmul simplex dual pentru aducerea la o formă explicită dual admisibilă și apoi pentru rezolvare. La sistemul restricțiilor (3.66) adăugăm restricția $x_2 + x_3 = M$, unde M este un număr pozitiv oricît de mare, iar noua variabilă x_3 trebuie să satisfacă condiția $x_3 \geqslant 0$. Prin pivotajul

	x_1	x_2	x_3	
f	0	−2	0	3
x_1	1	−1	0	−1
x_3	0	①	1	M
f	0	0	2	$2M+3$
x_1	1	0	1	$M-1$
x_2	0	1	1	M

obținem o formă explicită dual admisibilă a programuui auxiliar (P'). Aplicarea algoritmului simplex dual conduce la concluzia că programul (P') este rezolvat; totodată, variabila x_3 este variabilă secundară, deci sîntem în cazul d_2). Cum x_3 intră în expresia $f = -2M - 3 + 2x_3$ a funcției obiectiv, este vorba de subcazul d_2'). Concluzia dedusă în general este evidentă pe exemplul considerat: omițînd din soluția optimă găsită a programului (P') componenta x_3, rămîne $x_1 = M - 1$, $x_2 = M$, familie de soluții admisibile ale programului (P) pentru $M \in [1, +\infty)$; valorile corespunzătoare ale funcției obiectiv sînt $f = -2M - 3$. Funcția obiectiv are valori oricît de mici pentru $M \in [1, +\infty)$, deci programul liniar (P) nu are soluție optimă.

Exemplul 3.8. Considerăm programul liniar (P): să se minimizeze

$$f = 3 - x_2 + x_3, \tag{3.67}$$

cu restricţiile

$$x_1 + x_2 - x_3 = 2, \quad x_1 \geqslant 0, \; x_2 \geqslant 0, \; x_3 \geqslant 0. \tag{3.68}$$

Ne propunem să rezolvăm acest program liniar cu ajutorul metodei simplex duale, deşi programul se află deja sub o formă explicită primal admisibilă. La sistemul restricţiilor (3.68) adăugăm restricţia $x_2 + x_3 + x_4 = M$, unde M este un număr oricît de mare, iar noua variabila x_4 trebuie să satisfacă condiţia $x_4 \geqslant 0$. Prin pivotajul

	x_1	x_2	x_3	x_4	
f	0	−1	1	0	−3
x_1	1	1	−1	0	2
x_4	0	①	1	1	M
f	0	0	2	1	$M-3$
x_1	1	0	−2	−1	$-M+2$
x_2	0	1	1	1	M

obţinem o formă explicită dual admisibilă a programului auxiliar (P'). Aplicarea algoritmului simplex dual conduce la pivotajul

	x_1	x_2	x_3	x_4	
f	0	0	2	1	$M-3$
x_1	1	0	⊖2	−1	$-M+2$
x_2	0	1	1	1	M
f	1	0	0	0	−1
x_3	−1/2	0	1	1/2	$\frac{M}{2}-1$
x_2	1/2	1	0	1/2	$\frac{M}{2}+1$

care dă forma explicită finală a programului (P'). Variabila x_4 este variabilă secundară, deci sîntem în cazul d_2). Cum M nu intră în expresia $f = 1 + x_1$ a funcţiei obiectiv,

este vorba de cazul d_2'); concluzia dedusă în general este evidentă pe exemplul considerat: omiţînd din soluţia optimă găsită a programului (P') componenta x_4, rămîne $x_1 = 0,\ x_2 = \frac{M}{2} + 1,\ x_3 = \frac{M}{2} - 1$, familie de soluţii optime ale programului (P) pentru $M \in [2, +\infty)$; valoarea funcţiei obiectiv pentru toate soluţiile familiei este $f = 1$. Obţinem o soluţie optimă de bază a programului (P) pentru $M = 2$, anume $x_1 = 0,\ x_2 = 2,\ x_3 = 0$.

3.5. FORMELE REVIZUITE ALE ALGORITMILOR SIMPLECIALI

Se ştie din § 1.3 că un pivotaj asupra matricei

$$\bar{\mathscr{A}} = \left(\begin{array}{ccc|c} \dots & \alpha_0^h & \dots & -f_0 \\ \hline \dots & \alpha_{i_1}^h & \dots & \beta_{i_1} \\ & \vdots & & \vdots \\ \dots & \alpha_{i_m}^h & \dots & \beta_{i_m} \end{array}\right)$$

cu elementul pivot α_j^h se realizează prin înmulţirea lui $\bar{\mathscr{A}}$ la stînga cu o matrice de pivotaj, matrice care diferă de matricea unitate de ordinul $m+1$ printr-o singură coloană. Dacă α_j^h se găseşte în linia $s+1$ a matricei $\bar{\mathscr{A}}$, adică $j = i_s$, atunci coloana respectivă este coloana $s+1$ şi această coloană este dată de transpusul vectorului

$$(-\alpha_0^h/\alpha_j^h,\ -\alpha_{i_1}^h/\alpha_j^h,\ \dots,\ 1/\alpha_j^h,\ \dots,\ -\alpha_{i_m}^h/\alpha_j^h). \tag{3.69}$$

Astfel, pentru a se scrie matricea de pivotaj este suficient a se cunoaşte coloana pivotului. Se ştie din § 3.2 şi § 3.3 că operaţiile unui pas din algoritmul simplex primal sau din algoritmul simplex dual cer: examinarea liniei cu coeficienţii funcţiei obiectiv şi a coloanei termenilor liberi, apoi examinarea coloanei sau liniei pivotului. Aceste observaţii arată că pentru a se determina toate elementele necesare într-un pas al unuia dintre cei doi algoritmi trebuie să dispunem numai de o linie şi două coloane sau două linii şi o coloană din matricea formei explicite a programului liniar cunoscută în acel pas şi nu de matricea întreagă. Ideea centrală a formelor revizuite ale algoritmilor simpleciali este de a se păstra în tot timpul aplicării algoritmului matricea iniţială a problemei şi a se construi din ea, la fiecare pas, cu ajutorul vectorilor de forma

(3.69) obţinuţi în paşii precedenţi, toate elementele necesare desfăşurării pasului. Cum operaţiile unui pas pe baza elementelor cunoscute au fost descrise în § 3.2 şi § 3.3, rămîne să arătăm procedeul de construire a acestor elemente, direct din matricea iniţială. Desigur, avem de arătat cum se construieşte o linie şi o coloană dintr-o matrice asociată unei forme explicite, obţinută după r paşi ai algoritmului, $r > 0$.

Fie $M_1, \ldots, M_r$ matricele de pivotaj prin care s-au realizat r pivotaje succesive plecînd de la matricea $\bar{\mathcal{A}}$ corespunzătoare formei explicite iniţiale a unui program liniar, $(r > 0)$. Să notăm cu $\bar{\mathcal{A}}^r$ matricea rezultată după cele r pivotaje, adică

$$\bar{\mathcal{A}}^r = M_r \ldots M_1 \bar{\mathcal{A}}. \tag{3.70}$$

Linia i şi coloana k din $\bar{\mathcal{A}}^r$ se obţin matriceal din produsele

$$(\bar{\mathcal{A}}^r)_i = e_i' M_r \ldots M_1 \bar{\mathcal{A}}, \quad (\bar{\mathcal{A}}^r)^k = M_r \ldots M_1 (\bar{\mathcal{A}})^k, \tag{3.71}$$

unde e_i este coloana i din matricea unitate de ordinul $m + 1$, iar $(\bar{\mathcal{A}})^k$ reprezintă coloana k din $\bar{\mathcal{A}}$, Este vorba să ne ocupăm de particularităţile acestor produse.

Produsul $(\bar{\mathcal{A}}^r)_i$ se poate realiza sub forma

$$(\bar{\mathcal{A}}^r)_i = [(e_i' M_r) \ldots M_1] \bar{\mathcal{A}}, \tag{3.72}$$

cu alte cuvinte, făcînd de r ori calculul produsului dintre un vector linie şi o matrice de pivotaj; acest produs se efectuează pe larg astfel:

$$(b_0\ b_1 \ldots b_m) \begin{pmatrix} 1 & & & a_0 & & \\ & \ddots & & a_1 & & \\ & & \ddots & \vdots & & \\ & & & a_s & \ddots & \\ & & & \vdots & & \ddots \\ & & & a_m & & 1 \end{pmatrix} = (b_0\ b_1 \ldots \bar{b}_s \ldots b_m), \tag{3.73}$$

unde $\bar{b}_s = b_0 a_0 + b_1 a_1 + \ldots + b_m a_m$. Aşadar, produsul se realizează după următoarea regulă simplă: dacă matricea de pivotaj conţine în coloana $s + 1$ vectorul a, atunci după înmulţirea la

stînga a matricei cu un vector linie b se obţine un vector linie ce diferă de b numai prin componenta b_s înlocuită prin produsul scalar $\bar{b}_s = ab$. Desigur, ultima înmulţire din produsul (3.72) este înmulţirea dintre un vector linie şi matricea $\bar{\mathcal{A}}$.

Produsul $(\bar{\mathcal{A}}^r)^k$ se poate realiza sub forma

$$(\bar{\mathcal{A}}^r)^k = [M_r \ldots (M_1(\bar{\mathcal{A}})^k)], \tag{3.74}$$

cu alte cuvinte, făcînd de r ori produsul dintre o matrice de pivotaj şi un vector coloană; acest ultim produs se efectuează pe larg astfel:

$$\begin{pmatrix} 1 & & a_0 & & \\ & \ddots & a_1 & & \\ & & \ddots & \vdots & \\ & & & a_s & \ddots \\ & & & \vdots & & \ddots \\ & & & a_m & & & 1 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ b_s \\ \vdots \\ b_m \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ \vdots \\ 0 \\ \vdots \\ b_m \end{pmatrix} + b_s \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_s \\ \vdots \\ a_m \end{pmatrix}. \tag{3.75}$$

Aşadar, produsul se realizează după următoarea regulă simplă: dacă matricea de pivotaj conţine în coloana $s+1$ vectorul a, atunci produsul în chestiune se obţine adunînd vectorul obţinut din b prin înlocuirea cu zero a componentei b_s, cu produsul $b_s a$.

Pe baza regulilor simple (3.73), (3.75) vom putea reconstrui în fiecare pas, plecînd de la matricea iniţială $\bar{\mathcal{A}}$, toate elementele necesare aplicării algoritmului simplex primalui sau algoritmului simplex dual. Avantajul acestui mod de a proceda este că se reţin în memoria operativă a calculatorului numai vectorii ce reprezintă matricele de pivotaj, iar matricea iniţială se reţine în memoria externă.

Exemplul 3.9. Să reconsiderăm programul liniar din exemplul 3.2. Matricea iniţială a programului liniar este

$$\bar{\mathcal{A}} = \left(\begin{array}{ccccc|c} -2 & 0 & -3 & 0 & 0 & -5 \\ \hline -1 & 1 & 1 & 0 & 0 & 1 \\ 2 & 0 & -1 & 1 & 0 & 2 \\ -1 & 0 & 1 & 0 & 1 & 3 \end{array}\right).$$

Cum dispunem de o formă explicită primal admisibilă a programului liniar, vom aplica algoritmul simplex primal. Pasul 1 cere un pivotaj cu elementul pivot $\alpha_2^3 = 1$. Matricea de pivotaj corespunzătoare M_1 este reținută sub forma vectorului $b_1 = (2; 3\ 1\ 1\ -1)$, a cărui construcție este cunoscută din § 1.3. Pasul 2 cere mai întîi să se cunoască linia zero din $\bar{\mathscr{A}}^1$, adică $(\bar{\mathscr{A}}^1)_0 = (e_0' M_1)\bar{\mathscr{A}}$. Pe baza lui (3.73) avem $e_0' M_1 = (1\ 3\ 0\ 0)$, vector ce înmulțește $\bar{\mathscr{A}}$ și dă $(\bar{\mathscr{A}}^1)_0 = (-5\ 3\ 0\ 0\ 0 \mid -2)$. Acum sînt necesare în continuare prima coloană și coloana termenilor liberi din $\bar{\mathscr{A}}^1$, adică $(\bar{\mathscr{A}}^1)^1 = M_1(\bar{\mathscr{A}})^1$ și $(\bar{\mathscr{A}}^1)^6 = M_1(\bar{\mathscr{A}})^6$. Pe baza lui (3.75) avem $(\bar{\mathscr{A}}^1)^1 = (-2 \mid 0\ 2\ -1)' - (3 \mid 1\ 1\ -1)' = (-5 \mid -1\ \textcircled{1}\ 0)'$ și $(\bar{\mathscr{A}}^1)^6 = (-5 \mid 0\ 2\ 3)' + (3 \mid 1\ 1\ -1)' = (-2 \mid 1\ 3\ 2)'$. Dispunem de un singur raport cu numitor pozitiv, anume 3/1, încît elementul pivot este cel încercuit; din coloana pivotului formăm vectorul $v_2 = (3; 5\ 1\ 1\ 0)$ ce determină matricea de pivotaj corespunzătoare M_2. Pasul 3 cere mai întîi să se cunoască linia zero din $\bar{\mathscr{A}}^2$, adică $(\bar{\mathscr{A}}^2)_0 = [(e_0' M_2)\ M_1]\bar{\mathscr{A}}$. Pe baza lui (3.73) avem $e_0' M_2 = (1\ 0\ 5\ 0)$ și $(e_0' M_2)\ M_1 = (1\ 8\ 5\ 0)$; ultimul vector înmulțește $\bar{\mathscr{A}}$ și dă $(\bar{\mathscr{A}}^2)_0 = (0\ 8\ 0\ 5\ 0 \mid 13)$. Este clar că programul liniar este rezolvat; mai trebuie calculată ccloana termenilor liberi din $\bar{\mathscr{A}}^2$, adică $(\bar{\mathscr{A}}^2)^6 = M_2[M_1(\bar{\mathscr{A}})^6]$. Avem $M_1(\bar{\mathscr{A}})^6 = (-2 \mid 1\ 3\ 2)'$, iar pe baza lui (3.75) găsim $(\bar{\mathscr{A}}^2)^6 = (-2 \mid 1\ 0\ 2)' + 3(5 \mid 1\ 1\ 0)' = (13 \mid 4\ 3\ 2)'$.

Evidența schimbărilor de variabile principale fiind păstrată, această coloană dă o soluție optimă a programului liniar, după cum urmează: $x_1 = 3$, $x_2 = 0$, $x_3 = 4$, $x_4 = 0$, $x_5 = 2$ și valoarea corespunzătoare a functiei obiectiv $f = -13$.

Exemplul 3.10. Să reconsiderăm programul liniar din exemplul 3.3. Matricea inițială a programului liniar este

$$\bar{\mathscr{B}} = \left(\begin{array}{ccccc|c} 0 & 1 & 0 & 2 & 3 & 5 \\ \hline 1 & 1 & 0 & -2 & 1 & -2 \\ 0 & -1 & 1 & 1 & 1 & -3 \end{array}\right).$$

Cum dispunem de o formă explicită dual admisibilă a programului liniar, vom aplica algoritmul simplex dual. Pasul 1 cere un pivotaj cu elementul pivot $\alpha_3^2 = -1$. Matricea de pivotaj corespunzătoare este reținută sub forma vectorului $u_1 = (3; 1\ 1\ -1)$. Pasul 2 cere mai întîi să se cunoască coloana termenilor liberi din $\bar{\mathscr{B}}^1$, adică $(\bar{\mathscr{B}}^1)^6 = M_1(\bar{\mathscr{B}})^6$. Pe baza lui (3.75) avem $(\bar{\mathscr{B}}^1)^6 = (5 \mid -2\ 0)' - 3(1 \mid 1\ -1)' = (2 \mid -5\ 3)'$. Acum sînt necesare linia zero și linia întîi din $\bar{\mathscr{B}}^1$, adică $(\bar{\mathscr{B}}^1)_0 = (e_0' M_1)\bar{\mathscr{B}}$ și $(\bar{\mathscr{B}}^1)_1 = (e_1' M_1)\bar{\mathscr{B}}$. Pe baza lui (3.73) avem $e_0' M_1 = (1\ 0\ 1)$, vector ce înmulțește $\bar{\mathscr{B}}$ și dă $(\bar{\mathscr{B}}^1)_0 = (0\ 0\ 1\ 3\ 2 \mid 2)$ și $e_1' M_1 = (0\ 1\ 1)$, vector ce înmulțește $\bar{\mathscr{B}}$ și dă $(\bar{\mathscr{B}}^1)_1 = (1\ 0\ 1\ -1\ 0 \mid -5)$. Dispunem de un singur raport cu numitor negativ, anume 3/(−1), încît elementul pivot este cel încercuit; acum trebuie să calculăm coloana pivotului, adică $(\bar{\mathscr{B}}^1)^4 = M_1(\bar{\mathscr{B}})^4$. Pe baza lui (3.75) avem $(\bar{\mathscr{B}}^1)^4 = (2\ -2\ 0)' + (1\ 1\ -1)' = (3\ -1\ -1)'$; din coloana pivotului formăm vectorul $u_2 = (2; 3\ -1\ -1)$, ce determină matricea de pivotaj corespunzătoare M_2. Pasul 3 cere mai întîi să se cunoască coloana termenilor liberi din $\bar{\mathscr{B}}^2$, adică $(\bar{\mathscr{B}}^2)^6 = M_2[M_1(\bar{\mathscr{B}})^6]$. Avem $M_1(\bar{\mathscr{B}})^6 = (2 \mid -5\ 3)'$, iar pe baza lui (3.75) găsim $(\bar{\mathscr{B}}^2)^6 = (2 \mid 0\ 3)' - 5(3 \mid -1\ -1)' = (-13 \mid 5\ 8)'$. Este clar că programul liniar considerat este rezolvat; evidența schimbărilor de variabile principale fiind păstrată, această coloană dă o soluție optimă a programului liniar, după cum urmează: $x_1 = 0$, $x_2 = 8$, $x_3 = 0$, $x_4 = 5$, $x_5 = 0$ și valoarea corepsunzătoare a funcției obiectiv $f = 13$.

Scanned with OKEN Scanner

3.6. CAZUL DE DEGENERARE

Să considerăm din nou un program liniar sub o formă explicită primal admisibilă : să se minimizeze

$$f = f_0 + \sum_{k \in K} \alpha_0^k x_k \qquad (3.76)$$

supusă la restricţiile

$$x_i + \sum_{k \in K} \alpha_i^k x_k = \beta_i, \quad (i \in I), \qquad (3.77)$$

şi

$$x_i \geqslant 0, \; x_k \geqslant 0, \; (i \in I, \; k \in K), \qquad (3.78)$$

unde I este mulţimea indicilor variabilelor principale şi K este mulţimea indicilor variabilelor secundare, iar $\beta_i \geqslant 0$, $(i \in I)$. S-a arătat în § 2.5 că dacă se execută un pivotaj primal cu elementul pivot α_j^h, atunci valoarea funcţiei obiectiv pentru noua soluţie de bază primal admisibilă este

$$\bar{f} = f_0 + \frac{\beta_j}{\alpha_j^h} \cdot \alpha_0^h, \qquad (3.79)$$

unde $\alpha_0^h < 0$, $\alpha_j^h > 0$ şi $\beta_j \geqslant 0$. De aici a rezultat că dacă soluţia de bază admisibilă de plecare $x_i = \beta_i$, $x_k = 0$, $(i \in I, \; k \in K)$, era nedegenerată, adică $\beta_i > 0$, $(i \in I)$, vom avea şi $\beta_j > 0$, de unde $\bar{f} < f_0$. Prin urmare, dacă soluţia de plecare era nedegenerată, se obţinea în mod cert o nouă soluţie de bază admisibilă. Pe baza acestei observaţii şi a faptului că numărul soluţiilor de bază este finit s-a obţinut teorema 3.4 care ne asigură convergenţa algoritmului simplex primal. Ne propunem acum să examinăm problema convergenţei algoritmului în cazul cînd sistemul restricţiilor are soluţii de bază degenerate. Pentru aceasta facem din nou apel la consideraţiile din § 2.5 care arătau faptul fundamental că nedegenerarea asigură construirea unei soluţii de bază noi.

Să presupunem că soluţia de bază admisibilă de plecare era degenerată, adică printre numerele β_i, $(i \in I)$, există cel puţin

Scanned with OKEN Scanner

unul nul. După determinarea variabilei secundare x_h care devenea principală, prin regula

$$\alpha_0^h = \min_{k \in K} \alpha_0^k, \tag{3.80}$$

urma determinarea variabilei principale x_j care devenea secundară prin regula

$$\frac{\beta_j}{\alpha_j^h} = \min_{i/a_i^h > 0} \frac{\beta_i}{\alpha_i^h}. \tag{3.81}$$

Dacă notăm cu

$$I^+ = \{i \mid i \in I,\ \alpha_i^h > 0\}, \quad I^- = \{i \mid i \in I,\ \alpha_i^h \leqslant 0\}, \tag{3.82}$$

şi examinăm cu atenţie regula (3.81), observăm că pot apărea două situaţii : a) $\beta_i > 0$ pentru toţi $i \in I^+$; b) există cel puţin un $i \in I^+$ astfel încît $\beta_i = 0$. În primul caz, deşi soluţia de bază de plecare era degenerată, aveam $\beta_j > 0$, căci $j \in I^+$, şi, prin urmare, formula (3.79) ne asigură $\bar{f} < f_0$, adică s-a construit în mod cert o nouă soluţie de bază admisibilă, în general nedegenerată. În cazul al doilea, regula (3.81) arată că $j \in I^-$ şi $\beta_j = 0$. Acum, formulele (2.45) cu $\beta_j = 0$ nu lasă nici o îndoială asupra faptului că, deşi s-a obţinut o nouă formă explicită primal admisibilă corespunzătoare unui nou grup de variabile principale, noua soluţie de bază admisibilă coincide cu soluţia de plecare ; desigur, nici funcţia obiectiv nu şi-a schimbat valoarea. Noua soluţie de bază fiind degenerată, la un nou pivotaj primal se poate prezenta iarăşi situaţia b).

Această discuţie arată că într-o succesiune de pivotaje primale în care apar soluţii de bază admisibile degenerate este posibil ca începînd cu o anumită soluţie de bază degenerată să obţinem ulterior aceeaşi soluţie şi, prin urmare, s-ar putea ca după un număr de paşi să revenim la o formă explicită deja întîlnită. Desigur, în acest caz algoritmul nu duce la rezolvarea programului liniar. Fenomenul de revenire la o formă explicită anterioară a programului liniar este denumit fenomenul de *ciclare*. S-a arătat, prin construire de exemple ce pot fi găsite în [1, 18], că apariţia ciclării este posibilă practic şi, în consecinţă, este necesară asigurarea convergenţii algoritmului simplex primal printr-un grup de reguli suplimentare, în vederea evitării ciclării. Consideraţiile precedente se

pot relua cu privire la algoritmul simplex dual, unde se relevă de asemenea posibilitatea apariţiei ciclării.

Ceea ce merită subliniat este faptul că, deşi probleme de programare liniară care posedă soluţii de bază degenerate primal sau dual apar în practică, fenomenul de ciclare a survenit extrem de rar în aplicarea metodei simplex primale sau duale. De aceea, tehnicile de evitare a ciclării nu sînt în general incluse în programele curente pentru calculator, evitîndu-se astfel o cheltuială de timp de calcul inutilă. Pentru studiul unor asemenea tehnici pot fi consultate, de exemplu, lucrările [1, 2, 14].

În încheierea capitolului relativ la metodele generale de rezolvare a programelor liniare, relevăm încă faptul ce rezultă din studiul aplicării acestor metode, că în urma rezolvării unui program liniar se poate întîlni numai una dintre următoarele trei situaţii: 1) programul nu are soluţii admisibile; 2) problema are o funcţie obiectiv nemărginită pe mulţimea soluţiilor admisibile; 3) programul are soluţie optimă.

Scanned with OKEN Scanner

Partea a doua

TEHNICI DE REZOLVARE PENTRU PROBLEME CONEXE

4

PROGRAME PARAMETRICE

4.1. GENERALITĂȚI

Fie problema $P(\lambda)$: să se minimizeze

$$f(X, \lambda) = c'(\lambda) X, \tag{4.1}$$

cu restricțiile

$$A(\lambda) \cdot X = b(\lambda) \tag{4.2}$$

și

$$X \geqslant 0, \tag{4.3}$$

pentru diferitele valori ale parametrului λ, situate pe un interval sau o reuniunea de intervale. Pentru fiecare valoare a lui λ problema $P(\lambda)$ este un program liniar ; prin urmare, prin problema considerată se cere rezolvarea unei familii de programe liniare, toate cele obținute pentru diferitele valori ale lui λ. Desigur, programele liniare conținute în familia $P(\lambda)$ sînt legate între ele prin faptul că elementele numerice ale programelor se obțin din elemente funcționale, dînd valori parametrului. De aceea, soluțiile optimale ale programelor familiei au anumite proprietăți, ce vor fi relevate în continuare ; aceste proprietăți permit rezolvarea familiei de programe liniare prin aplicarea unor cicluri de pași ale uneia dintre metodele simpleciale, primală sau duală. Semnalăm că (4.1) —(4.3) pot defini, împreună cu restricția de apartenență a parametrului la o anumită mulțime de valori, și un program neliniar, dar în acest caz s-ar cere mai puțin decît în problema formulată mai sus.

Problema $P(\lambda)$ este denumită *program liniar uniparametric*. Cele mai simple programe liniare uniparametrice sînt cele în care avem una dintre următoarele două situații :

Scanned with OKEN Scanner

a) A şi b sînt matrice numerice, iar $c(\lambda) = c^0 + \lambda c^1$, unde c^0 şi c^1 sînt vectori numerici;

b) A şi c sînt matrice numerice, iar $b(\lambda) = b^0 + \lambda b^1$, unde b^0, b^1 sînt vectori numerici.

Aceste două cazuri, pe care le vom studia în continuare, au fost şi primele tipuri de programe parametrice studiate în literatură (v. [6, 7]). Astăzi se cunoaşte rezolvarea numerică a cazului general uniparametric şi chiar şi a cazului multiparametric, acela cînd în (4.1) — (4.3) λ este un vector al parametrilor, în loc de un parametru scalar.

Programarea parametrică are importanţă practică, deoarece de multe ori prezintă interes cunoaşterea modului în care variază soluţia optimă a unui program liniar, odată cu variaţia unuia sau mai multor elemente date ale programului. Aceste elemente fiind resurse, costuri, productivităţi sau alte mărimi economice, putem înlocui unul dintre elemente sau mai multe cu unul sau respectiv mai mulţi parametri şi apoi stabili, prin rezolvarea programului parametric, comportarea unei soluţii optimale, cînd parametrii variază în domeniul valorilor pe care practica le admite. Desigur, dificultatea rezolvării depinde atît de numărul parametrilor, cît şi de modul în care intră aceşti parametri în datele problemei.

Exemplul 4.1. Considerăm problema: o maşină poate produce două repere, A şi B, cu productivităţile orare respective de 50 piese şi 75 piese; ştiind că maşina poate lucra 20 ore săptămînal şi că dispunem de capacităţi de stocaj respective de 500 piese şi 1000 piese, se pune problema de a se stabili planul de lucru săptămînal cu rentabilitate maximă, beneficiul unitar fiind pe reper de 2 unităţi şi respectiv 1 unitate valorică.

Definim necunoscutele: x = numărul pieselor de tip A, y = numărul pieselor de tip B, de produs săptămînal în cadrul planului optim. Restricţiile pentru aceste variabile sînt de două tipuri, restricţia de timp de lucru pentru maşină şi restricţiile de stocaj pentru cele două produse. Timpul de lucru consumat pentru producţia reperului A este $x/50$ ore, iar timpul de lucru consumat pentru producţia reperului B este $y/75$ ore, încît timpul de lucru consumat pentru ambele este $x/50 + y/75$ ore. Atunci, restricţiile problemei se scriu

$$\left.\begin{array}{l} \dfrac{x}{50} + \dfrac{y}{75} \leqslant 20 \\ x \leqslant 500,\ y \leqslant 1\,000 \end{array}\right\}. \tag{4.4}$$

La acestea trebuie adăugate condiţiile de nenegativitate

$$x \geqslant 0, \quad y \geqslant 0, \tag{4.5}$$

impuse de sensul economic al variabilelor. Beneficiul total realizat săptămînal este dat de funcţia

$$f = 2x + y, \tag{4.6}$$

şi deci problema considerată se formulează : să se maximizeze funcţia (4.6), cu restricţiile (4.4) şi (4.5). Este clar că avem de-a face cu un program liniar obişnuit, ce poate fi rezolvat printr-una din metodele cunoscute. Să admitem însă că ar apărea posibilitatea de a se spori timpul de lucru săptămînal al maşinii, de exemplu prin costuri săptămînale suplimentare, şi s-ar pune problema de a se stabili dacă se realizează efectiv un beneficiu suplimentar pe această cale. Să notăm cu k costul de prelungire cu o oră a timpului de lucru săptămînal al maşinii şi cu λ numărul de ore suplimentare săptămînale ale maşinii ; k se presupune constant pentru $\lambda \in [0, 20]$, iar prelungirea timpului de lucru peste 40 ore săptămînal se presupune imposibilă. Atunci, ultima problemă formulată cere mai întîi de a se maximiza funcţia (4.6) în condiţiile

$$\left.\begin{aligned} \frac{x}{50} + \frac{y}{75} &\leqslant 20 + \lambda \\ x \leqslant 500, \quad y &\leqslant 1\,000 \end{aligned}\right\} \tag{4.7}$$

şi condiţiile de nenegativitate (4.5), λ fiind un parametru cu valori în intervalul $[0, 20]$; desigur, odată cu o soluţie optimă a programului parametric se va obţine şi valoarea corespunzătoare a beneficiului ca funcţie de λ. Apoi, problema considerată impune a se face diferenţa dintre această ultimă expresie a beneficiului şi costul de prelungire a timpului de lucru $k\lambda$ adunat cu beneficiul realizat pentru $\lambda = 0$, obţinîndu-se astfel beneficiul suplimentar săptămînal ca funcţie de λ. Examinînd această funcţie, vom putea stabili dacă este cazul a se prelungi timpul de lucru săptămînal al maşini şi cu cît anume. Aşa cum vom constata ulterior, se va obţine o funcţie liniară pe porţiuni, astfel încît va fi simplu a se stabili rezultatul cerut.

Rezolvarea programelor liniare parametrice de tipul a) şi b) cu ajutorul algoritmului simplex primal şi, respectiv, dual se reduce la rezolvarea unui număr finit de programe liniare obişnuite depinzînd de o nedeterminată cu valori foarte mici şi pozitive. De aceea, înainte de a trece la prezentarea procedeelor de rezolvare a programelor parametrice de tipul a) şi b), vom expune de fiecare dată aspecte ale algoritmului simplex primal şi respectiv dual, adaptate pentru rezolvarea acestor programe liniare obişnuite cu o nedeterminată.

4.2. PROGRAME LINIARE DE FORMĂ EXPLICITĂ PRIMAL ADMISIBILĂ CU O NEDETERMINATĂ ÎN FUNCŢIA OBIECTIV

Considerăm programul liniar $\pi(\nu)$: să se minimizeze

$$\varphi(X, \nu) = \sigma_0 + \nu\tau_0 + \sum_{k \in K} (\sigma_0^k + \nu\tau_0^k)\, x_k, \tag{4.8}$$

cu restricțiile

$$x_i + \sum_{k \in K} \sigma_i^k x_k = \rho_i, \quad (i \in I), \tag{4.9}$$

și

$$x_i \geqslant 0, \; x_k \geqslant 0, \; (i \in I, \; k \in K), \tag{4.10}$$

unde ν este un număr fixat pozitiv oricît de mic și

$$\rho_i \geqslant 0, \quad (i \in I). \tag{4.11}$$

Deoarece $\pi(\nu)$ este un program liniar obișnut de formă explicită primal admisibilă, pentru rezolvare vom aplica algoritmul simplex primal. Se știe (v. § 3.2) că operațiile unui pas al algoritmului sînt: aplicarea criteriului de optimalitate, apoi dacă este necesară aplicarea criteriului de intrare, apoi dacă este necesară aplicarea criteriului de ieșire și trecerea la o nouă formă explicită primal admisibilă. În continuare, ne interesează care sînt particularitățile acestor operații, datorate prezenței numărului nedeterminat ν.

Să presupunem că au fost realizați deja r pași ai algoritmului simplex primal, $(r \geqslant 0)$, pentru rezolvarea programului $\pi(\nu)$. Ca să nu mai complicăm notațiile vom admite că (4.8) — (4.10) este forma explicită primal admisibilă a lui $\pi(\nu)$ obținută după cei r pași, și vom trece la descrierea operațiilor pasului $r+1$.

La aplicarea criteriului de optimalitate ne interesează să stabilim semnul unei expresii de forma $\sigma_0^k + \nu\tau_0^k$, cînd ν are o valoare pozitivă oricît de mică. Este clar că semnul dorit este semnul lui σ_0^k, dacă $\sigma_0^k \neq 0$, și semnul lui τ_0^k, dacă $\sigma_0^k = 0$ și $\tau_0^k \neq 0$. De aceea, criteriul de optimalitate

$$\sigma_0^k + \nu\tau_0^k \geqslant 0, \; (k \in K), \tag{4.12}$$

se transcrie sub forma

$$\begin{gathered} \sigma_0^k \geqslant 0, \; (k \in K), \\ \sigma_0^k = 0 \Rightarrow \tau_0^k \geqslant 0. \end{gathered} \tag{4.13}$$

Dacă am scrie coeficienții funcției obiectiv în matricea cu care lucrăm la rezolvarea lui $\pi(\nu)$ pe două linii în loc de una, folosind

prima linie pentru termenii liberi ai coeficienţilor şi cealaltă pentru coeficienţii lui ν, atunci forma (4.13) a criteriului se traduce astfel: elementele primei linii trebuie să fie nenegative, iar elementele celeilalte corespunzătoare elementelor nule din prima linie trebuie de asemenea să fie nenegative. Desigur, criteriul (4.13) stabileşte dacă soluţia de bază

$$x_i = \rho_i, \; x_k = 0, \; (i \in I, \; k \in K), \tag{4.14}$$

este optimă, sau trebuie să trecem la operaţia următoare. În primul caz, obţinem şi valoarea funcţiei obiectiv $f = \sigma_0 + \nu\tau_0$, care depinde de ν, deşi soluţia (4.14) nu depinde de nedeterminată.

La aplicarea criteriului de intrare ne interesează a compara expresiile $\sigma_0^k + \nu\tau_0^k$, $(k \in K)$, pentru a găsi pe cea mai mică dintre ele. Este clar că dintre două expresii $\sigma + \nu\tau$ şi $\sigma' + \nu\tau'$, în care ν are o valoare pozitivă oricît de mică, aceea cu termenul liber mai mic va fi mai mică, iar dacă termenii liberi sînt egali, accea cu coeficientul lui ν mai mic va fi mai mică. De aceea, criteriul de intrare

$$\sigma_0^h + \nu\tau_0^h = \min_k \; (\sigma_0^k + \nu\tau_0^k) \tag{4.15}$$

se poate formula astfel: fie

$$L = \{p \mid \sigma_0^p = \min_k \; \sigma_0^k\}; \tag{4.16}$$

dacă $|L| = 1$, atunci x_h cu $h \in L$ va deveni variabilă principală, iar dacă $|L| > 1$, atunci o variabilă x_h, cu h determinat prin

$$\tau_0^h = \min_{p \in L} \; \tau_0^p, \tag{4.17}$$

va deveni principală (cu $|L|$ s-a notat numărul elementelor mulţimii L). Bineînţeles, ar fi posibil ca nici condiţia (4.17) să nu determine în mod unic variabila ce devine principală, dar am precizat că în lipsa unicităţii se va alege o variabilă dintre cele ce satisfac (4.17). Dacă am întrebuinţa scrierea coeficienţilor funcţiei obiectiv pe două linii în matricea formei explicite, criteriul de intrare s-ar traduce astfel: se determină cel mai mic element al primei linii, sau dacă există mai multe cu valoarea minimă, cel mai mic element al celeilalte linii din coloanele corespunzătoare elementelor cu valoa-

Scanned with OKEN Scanner

rea minimă din prima linie. Desigur, după aplicarea criteriului de intrare (4.15) va trebui să cercetăm elementele din coloana variabilei x_h pentru a stabili fie că programul liniar nu are soluție optimă, fie că este necesar să trecem la operația următoare.

La aplicarea criteriului de ieșire elementele utilizate sînt elemente ale sistemului și cum acestea nu depind de ν, criteriul de ieșire va avea aspectul obișnuit și va determina variabila x_j ce devine secundară.

Pivotajul cu elementul pivot σ_{jj}^{h}, prin care se realizează trecerea la o nouă formă explicită primal admisibilă, se realizează de asemenea sub forma obișnuită, căci se lucrează cu elementele liniei j ce nu depind de ν. Se înțelege, pentru eliminarea variabilei x_k din funcția obiectiv va trebui să înmulțim linia j cu o expresie liniară în ν și s-o adunăm la linia funcției obiectiv, sau atunci cînd două linii conțin elementele funcției obiectiv va trebui să le tratăm ca pe oricare dintre celelalte linii ale matricei.

Chiar și fără aplicarea unei reguli de evitare a ciclării, în general, după un număr finit de pași ai algoritmului simplex primal programul $\pi(\nu)$ va fi rezolvat. Remarcăm că apariția alternativelor ce decid oprirea algoritmului depinde de valoarea lui ν. De asemenea valoarea optimă depinde de ν, în timp ce soluția optimă nu depinde de nedeterminată.

Exemplul 4.2. Considerăm programul $\pi(\nu)$: să se minimizeze

$$\varphi(x, \nu) = 2 + \nu + (\nu - 3)\, x_4 + (2\nu - 1)\, x_5, \tag{4.18}$$

cu restricțiile

$$\left.\begin{aligned} x_1 \quad + x_4 + x_5 &= 1 \\ x_2 \quad + x_4 - x_5 &= 2 \\ x_3 + x_4 + 2x_5 &= 3 \end{aligned}\right\} \tag{4.19}$$

și

$$x_1 \geqslant 0, \ldots, x_5 \geqslant 0, \tag{4.20}$$

unde ν este un număr pozitiv oricît de mic. Matricea asociată formei explicite primal admisibile disponibile este

$$\overline{\mathscr{A}} = \left(\begin{array}{ccccc|c} 0 & 0 & 0 & -3 & -1 & -2 \\ 0 & 0 & 0 & 1 & 2 & -1 \\ \hline 1 & 0 & 0 & \textcircled{1} & 1 & 1 \\ 0 & 1 & 0 & 1 & -1 & 2 \\ 0 & 0 & 1 & 1 & 2 & 3 \end{array}\right).$$

Elementul minim al primei linii este negativ și corespunde variabilei x_4 ce va deveni principală; raportul minim cerut de criteriul de ieșire este dat de ecuația cu variabila principală x_1, variabilă care va deveni secundară. Pivotajul cu elementul pivot $\sigma_1^4 = 1$ conduce la matricea

$$\bar{\mathscr{A}}^1 = \left(\begin{array}{ccccc|c} 3 & 0 & 0 & 0 & 2 & 1 \\ -1 & 0 & 0 & 0 & 1 & -2 \\ \hline 1 & 0 & 0 & 1 & 1 & 1 \\ -1 & 1 & 0 & 0 & -2 & 1 \\ -1 & 0 & 1 & 0 & 1 & 2 \end{array}\right).$$

Elementul minim al primei linii este nul și elementul minim dintre elementele liniei a doua situate în coloanele cu elemente nule în prima linie este tot nul. Soluția de bază $x_1 = 0$, $x_2 = 1$, $x_3 = 2$, $x_4 = 1$, $x_5 = 0$ este o soluție optimă a programului $\pi(\nu)$, iar valoarea corespunzătoare a funcției obiectiv este $f = 2\nu - 1$.

4.3. PROGRAME LINIARE PARAMETRICE CU PARAMETRUL ÎN FUNCȚIA OBIECTIV

Considerăm un program liniar parametric $P(\lambda)$ de forma a): să se minimizeze

$$f(x, \lambda) = (c^0 + \lambda c^1) X, \tag{4.21}$$

în condițiile

$$AX = b, \; X \geqslant 0, \tag{4.22}$$

unde λ este un parametru cu valori într-un interval $\mathcal{J} = [0, \sigma)$, σ fiind un număr pozitiv sau $+\infty$.

În prealabil să arătăm că, din punctul de vedere al rezolvării programului, ne putem limita la prezentarea cazului cînd programul este deja sub forma explicită primal admisibilă. Pentru aceasta să admitem că sistemul $AX = b$ n-ar fi explicit, sau termenii liberi n-ar fi toți nenegativi; atunci, am putea proceda exact la fel ca în cazul cînd s-ar cere a se aduce un program liniar obișnuit la forma explicită, deoarece pentru aceasta nu se folosește funcția $f(X, \lambda)$ în care intră parametrul. Cu alte cuvinte, procedeul de aducere la forma explicită primal admisibilă a programului (4.21), (4.22) este următorul: după ce ne-am asigurat nenegativitatea termenilor liberi, înmulțind eventual unele ecuații cu -1, vom introduce în fiecare ecuație o variabilă artificială și vom minimiza

suma variabilelor artificiale; drept rezultat fie găsim că sistemul liniar $AX = b$ nu are soluţii admisibile şi, prin urmare, nici programul liniar parametric nu are soluţii admisibile pentru nici un $\lambda \in [0, \sigma)$, fie găsim o formă explicită a sistemului $AX = b$ cu toţi termenii liberi nenegativi. În ultimul caz, după eliminarea variabilelor principale din funcţia obiectiv vom găsi o formă explicită primal admisibilă a programului liniar parametric. Desigur, după eliminare, coeficienţii variabilelor şi termenul liber din funcţia obiectiv sînt tot funcţii liniare de λ.

Să considerăm că operaţiile de aducere la o formă explicită primal admisibilă a programului $P(\lambda)$ au fost realizate şi să trecem la descrierea rezolvării programului.

Rezolvarea programului liniar parametric $P(\lambda)$ se va realiza printr-un număr finit de grupuri de paşi ai algoritmului simplex primal, fiecare grup de paşi fiind aplicat unui program liniar parametric echivalent lui $P(\lambda)$ în baza unor schimbări de parametri şi a unor pivotaje succesive efectuate în grupurile de paşi anterioare. După cum se va arăta în cele ce urmează, orice grup de paşi rezolvă problema $P(\lambda)$ pe un subinterval al intervalului $[0, \sigma)$.

Să admitem că s-au executat s grupuri de paşi, $s \geqslant 0$. Dacă $s > 0$, să presupunem că pe această cale s-a realizat rezolvarea lui $P(\lambda)$ pe intervalul $\mathcal{J}_s = [0, \lambda^s)$, unde $0 < \lambda^s < \sigma$; să presupunem de asemenea că în baza schimbării de parametru $\lambda = \lambda^s + \nu$ şi a pivotajelor primale realizate în grupurile de paşi anterioare dispunem de un program liniar parametric $\pi(\nu)$ de formă explicită primal admisibilă. Atunci, înainte de a trece la grupul de paşi $s + 1$, dispunem oricare ar fi $s \geqslant 0$ de un program $\pi(\nu)$ de formă explicită primal admisibilă, astfel ca $\pi(\lambda - \lambda^s)$ este echivalent cu $P(\lambda)$, $(\lambda^0 = 0)$. Programul $\pi(\nu)$ urmează a fi rezolvat pe intervalul $[0, \sigma^s)$, $\sigma^s = \sigma - \lambda^s$, căci $P(\lambda)$ trebuie rezolvat încă pe intervalul $[\lambda^s, \sigma)$. Să trecem acum la descrierea operaţiilor mari ale grupului de paşi $s + 1$.

(A) Se rezolvă programul liniar ordinar obţinut din programul liniar parametric $\pi(\nu)$ atunci cînd se consideră parametrul ν fixat la o valoare pozitivă oricît de mică. Rezolvarea se face prin algoritmul simplex primal adaptat sub forma dată în paragraful precedent. Drept rezultat se obţine fie o soluţie de bază optimă (S^{s+1}) a acestui program liniar ordinar, fie concluzia că funcţia obiectiv a sa poate lua valori oricît de mici.

Scanned with OKEN Scanner

(B) Se determină un interval $(0, \nu^*)$ cu proprietatea că pentru orice $\nu \in (0, \nu^*)$ programul $\pi(\nu)$ are aceeaşi rezolvare ca pentru ν pozitiv oricît de mic; cu alte cuvinte, pentru orice $\nu \in (0, \nu^*)$, fie (S^{s+1}) este o soluţie optimă a lui $\pi(\nu)$, fie acest program nu are soluţie optimă, după cum aceeaşi alternativă a fost stabilită cînd ν era un număr pozitiv fixat oricît de mic.

Acum, dacă $\nu^* \geqslant \sigma^s$, programul liniar parametric $\pi(\nu)$ este rezolvat pe intervalul $(0, \sigma^s)$; se va vedea că adesea dispunem şi de soluţia lui pentru $\nu = 0$. De aceea, în acest caz, rezolvarea programului $P(\lambda)$ este încheiată. Dacă însă $\nu^* < \sigma^s$, programul liniar parametric $\pi(\nu)$ este rezolvat pe intervalul $(0, \nu^*) \subset [0, \sigma^s)$; se va vedea că dispunem şi de o soluţie optimă pentru $\nu = 0$. Deci, rezolvarea programului $P(\lambda)$ este încheiată pe intervalul $\mathcal{I}_{s+1} = [0, \lambda^{s+1})$, $\lambda^{s+1} = \lambda^s + \nu^*$, dar trebuie să continue pe intervalul $[\lambda^{s+1}, \sigma)$.

(C) Dacă $\nu^* < \sigma^s$, se face în forma explicită primal admisibilă finală a lui $\pi(\nu)$ schimbarea de parametru $\nu = \nu^* + \theta$, unde θ este un nou parametru. Prin aceasta se obţie un program liniar parametric echivalent cu $P(\lambda)$ în baza schimbării de parametru $\lambda = \lambda^{s+1} + \theta$ şi a pivotajelor celor $s+1$ grupuri de paşi. Noul program parametric este sub o formă explicită primal admisibilă şi urmează a fi rezolvat pentru $\theta \in [0, \sigma^{s+1})$, $\sigma^{s+1} = \sigma - \lambda^{s+1}$. Sîntem pregătiţi pentru a trece la grupul de paşi $s+2$.

Deşi procedeul de rezolvare este deja precizat, să trecem la analiza detaliată a operaţiilor (A), (B), (C), în vederea formulării exacte a rezultatelor grupului de paşi $s+1$.

(A) Să admitem că forma explicită primal admisibilă finală obţinută prin aplicarea operaţiei (A) asupra programului $\pi(\nu)$ este următoarea: să se minimizeze

$$\varphi(X, \nu) = \sigma_0 + \nu\tau_0 + \sum_{k \in K} (\sigma_0^k + \nu\tau_0^k)\, x_k, \tag{4.23}$$

cu restricţiile

$$x_i + \sum_{k \in K} \sigma_i^k x_k = \rho_i, \quad (i \in I), \tag{4.24}$$

şi

$$x_i \geqslant 0, \; x_k \geqslant 0, \quad (i \in I, \; k \in K). \tag{4.25}$$

Scanned with OKEN Scanner

În (4.24) avem $\rho_i \geqslant 0$, $(i \in I)$, și întrucît (4.23) — (4.25) este forma explicită primal admisibilă finală, obținută după operația (A), am putut întîlni una dintre alternativele:

(A_1) $\sigma_0^k + \nu\tau_0^k \geqslant 0$, $(k \in K)$, pentru ν pozitiv oricît de mic;

(A_2) $\sigma_0^h + \nu\tau_0^h < 0$, pentru ν pozitiv oricît de mic și un anumit $h \in K$, iar $\sigma_i^h \leqslant 0$, $(i \in I)$.

În cazul (A_1) soluția de bază

$$(S^{s+1}) \quad x_i = \rho_i, \quad x_k = 0, \quad (i \in I,\ k \in K), \tag{4.26}$$

este o soluție optimă a programului $\pi(\nu)$, iar în cazul (A_2) programul $\pi(\nu)$ nu are soluții optime, pentru ν pozitiv oricît de mic. Va trebui să ne ocupăm în continuare de ambele alternative, pentru a stabili care poate fi rezultatul operației (B).

(B) Acum avem de determinat un interval cît mai mare, pe care situația (A_1) sau (A_2), stabilită pentru o valoare pozitivă oricît de mică a lui ν, să se mențină.

(B_1) Dacă s-a obținut alternativa (A_1), înseamnă că sistemul de inecuații liniare

$$\sigma_0^k + \nu\tau_0^k \geqslant 0, \ (k \in K), \tag{4.27}$$

care asigură optimalitatea soluției de bază (S^{s+1}), și care este compatibil pentru ν pozitiv oricît de mic, este format din inecuații de unul sau mai multe dintre următoarele trei tipuri:

$$\sigma_0^k + \nu\tau_0^k \geqslant 0, \ (\sigma_0^k > 0,\ \tau_0^k \geqslant 0); \tag{4.28}$$

$$\nu\tau_0^k \geqslant 0, \ (\tau_0^k \geqslant 0); \tag{4.29}$$

$$\sigma_0^k + \nu\tau_0^k \geqslant 0, \ (\sigma_0^k > 0,\ \tau_0^k < 0). \tag{4.30}$$

Definim ν^* astfel:

$$\nu^* = \begin{cases} \min\limits_{k \in K'} \left(-\dfrac{\sigma_0^k}{\tau_0^k} \right), & K' \neq \emptyset, \\ +\infty, & K' \neq \emptyset \end{cases} \tag{4.31}$$

unde

$$K' = \{k \mid k \in K,\ \sigma_0^k > 0,\ \tau_0^k < 0\}. \tag{4.32}$$

Este clar că inecuațiile (4.27) sînt satisfăcute pentru orice $\nu \in [0, \nu^*)$ și, dacă $\nu^* < +\infty$, cel puțin una dintre inecuații nu este satisfăcută pentru $\nu > \nu^*$; prin urmare, soluția de bază (S^{s+1}) este o soluție optimă a programului $\pi(\nu)$ pe intervalul $[0, \nu^*)$ și dacă $\nu^* < +\infty$, nu este optimă pe $(\nu^*, +\infty)$. Observăm că (S^{s+1}) este soluția optimă și pentru $\nu = \nu^*$, cînd $\nu^* < +\infty$. Programele $P(\lambda)$ și $\pi(\nu)$ sînt rezolvate pe intervalele corespunzătoare $[\lambda^s, \lambda^s + \nu^*)$ și $[0, \nu^*)$; desigur, valorile optime corespunzătoare se exprimă diferit: $\sigma_0 + (\lambda - \lambda^s)\tau_0$ și $\sigma_0 + \nu\tau_0$.

(B_2) Dacă s-a obținut alternativa (A_2), înseamnă că inecuația

$$\sigma_0^h + \nu\tau_0^h < 0 \tag{4.33}$$

este satisfăcută pentru ν pozitiv oricît de mic; prin urmare, inecuația (4.33) este de unul dintre următoarele trei tipuri:

$$\sigma_0^h + \nu\tau_0^h < 0, \qquad (\sigma_0^h < 0,\ \tau_0^h \leqslant 0); \tag{4.34}$$

$$\nu\tau_0^h < 0, \qquad (\tau_0^h < 0); \tag{4.35}$$

$$\sigma_0^h + \nu\tau_0^h < 0, \qquad (\sigma_0^h < 0,\ \tau_0^h > 0). \tag{4.36}$$

Definim ν^* prin

$$\nu^* = \begin{cases} -\dfrac{\sigma_0^h}{\tau_0^h} \text{ cînd } \sigma_0^h < 0,\ \tau_0^h > 0, \\ +\infty \text{ în celelalte cazuri.} \end{cases} \tag{4.37}$$

Este clar că inecuația (4.33) este satisfăcută pentru orice $\nu \in (0, \nu^*)$ și dacă $\nu^* < +\infty$ nu este satisfăcută pentru $\nu \geqslant \nu^*$; prin urmare, programul $\pi(\nu)$ nu are soluții optime pe intervalul $(0, \nu^*)$. Obținem că pentru $\nu = 0$ programul ar putea avea soluții optime numai dacă inecuația (4.33) este de forma (4.35). Programele $P(\lambda)$ și $\pi(\nu)$ sînt rezolvate pe intervalele corespunzătoare $(\lambda^s, \lambda^s + \nu^*)$ și $(0, \nu^*)$; oricum intervalul pe care rezultatul obținut pentru ν pozitiv oricît de mic se menține este o semidreaptă, fie este vorba de semidreapta $(-\infty, \nu^*)$, fie de o semidreaptă care include $(0, +\infty)$.

Considerațiile precedente demonstrează teorema următoare:

Teorema 4.1. *Fiind dat un program liniar parametric* $\pi(\nu)$ *sub o formă explicită primal admisibilă, se poate determina* ν^*, $\nu^* \in (0, +\infty)$, *astfel ca:*

— *sau programul are o soluție optimă pe intervalul* $[0, \nu^*)$, *care pentru* $\nu^* < +\infty$ *este optimă chiar pe intervalul închis* $[0, \nu^*]$ *și nu este optimă pentru* $\nu > \nu^*$;

— *sau programul nu are soluții optime pe intervalul* $(0, \nu^*)$, *situație care se menține pentru* $\nu = 0$ *și nu are loc pentru* $\nu \geqslant \nu^*$ *cînd* $\nu^* < +\infty$.

(C). Dacă $\nu^* \geqslant \sigma^s$, programul parametric $P(\lambda)$ este rezolvat pe intervalul $[0, \sigma)$, dacă însă $\nu^* < \sigma^s$, este clar din teorema 4.1 că în ambele cazuri (A_1) și (A_2) programul parametric $P(\lambda)$ este rezolvat pe intervalul $[0, \lambda^{s+1})$, $\lambda^{s+1} = \lambda^s + \nu^*$, ba chiar în primul caz și pentru $\lambda = \lambda^{s+1}$. Schimbarea de parametru $\nu = \nu^* + \theta$ ne conduce la un program parametric sub o formă explicită primal admisibilă, echivalent lui $P(\lambda)$. Procedeul se reia cu un nou grup de pași.

Din teorema 4.1, din faptul că numărul soluțiilor de bază ale sistemului (4.22) este finit și din convergența algoritmului simplex primal rezultă convergența algoritmului simplex primal adaptat pentru rezolvarea programelor parametrice de tipul (a). Situația finală obținută în urma rezolvării programului $P(\lambda)$ prin $p+1$ grupuri de pași, $(p \geqslant 0)$, poate fi una dintre următoarele:

$p = 0$ pentru $\lambda \in (0, \sigma)$ fie s-a găsit o soluție optimă (S^1), fie s-a stabilit inexistența soluțiilor optime; în primul caz, (S^1) este soluție optimă și pentru $\lambda = 0$, în timp ce în cazul al doilea valoarea $\lambda = 0$ în general trebuie examinată separat;

$p > 0$: s-au determinat numerele $\lambda^1, \lambda^2, \ldots, \lambda^p$, $0 < \lambda^1 < \lambda^2 < \ldots < \lambda^p < \sigma$, și soluțiile de bază $(S^1), (S^2), \ldots, (S^p)$, astfel încît (S^{s+1}) este o soluție optimă a programului $P(\lambda)$ pentru orice $\lambda \in [\lambda^s, \lambda^{s+1}]$, $(s = 0, 1, \ldots, p-1)$, iar pentru $\lambda \in (\lambda^p, \sigma)$ fie s-a găsit o soluție de bază optimă (S^{p+1}), fie s-a stabilit inexistența soluțiilor optime.

Remarcăm că oricare dintre valorile λ^s care este extremitate comună la două intervale vecine, unde programul $P(\lambda)$ are soluții optime, este o valoare a parametrului pentru care programul are măcar două soluții de bază optime diferite, anume (S^s) și (S^{s+1}); putem verifica imediat că în acest caz

$$X = \alpha X^s + (1 - \alpha) X^{s+1} \tag{4.38}$$

reprezintă de asemenea o soluție optimă oricare ar fi α, $0 \leqslant \alpha \leqslant 1$, unde X^s și X^{s+1} sînt cele două soluții optime. Numerele $\lambda^1, \lambda^2, \ldots, \lambda^p$ se numesc *valori critice* ale programului liniar parametric

Scanned with OKEN Scanner

$P(\lambda)$; cunoaşterea acestor numere are o importanţă practică deosebită.

Ne-am ocupat pînă acum de problema rezolvării unui program liniar parametric $P(\lambda)$ pe un interval de forma $[0, \sigma)$, $\sigma \in (0, +\infty)$. Dacă intervalul pe care se cere a se rezolva programul $P(\lambda)$ este de forma $[\rho, \sigma)$, unde ρ este un număr real oarecare, schimbarea de parametru $\lambda = \rho + \mu$ va transforma intervalul considerat în intervalul $[0, \sigma - \rho)$. Dacă intervalul pe care se cere a se rezolva programul $P(\lambda)$ este de forma $(-\infty, +\infty)$, rezolvarea se poate face în două etape, mai întîi pe intervalul $[0, +\infty)$ şi apoi pe intervalul $(-\infty, 0)$ cu schimbarea de parametru prealabilă $\lambda = -\mu$.

Exemplul 4.3. Considerăm programul liniar parametric $\mathcal{P}(\lambda)$: să se maximizeze

$$f(x, y, \lambda) = \lambda x + y, \tag{4.39}$$

cu restricţiile

$$\frac{x}{50} + \frac{y}{75} \leqslant 20, \; x \leqslant 500, \; y \leqslant 1000, \tag{4.40}$$

şi

$$x \geqslant 0, \; y \geqslant 0, \tag{4.41}$$

unde $\lambda \in [0, +\infty)$.

Pentru $\lambda = 2$ se obţine programul liniar din exemplul 4.1; totuşi după cum se va vedea, prezintă interes, un studiu parametric. Semnificaţia parametrului λ este evidentă: beneficiul unitar la reperul A. Problema practică din exemplul 4.1 s-a transformat în problema mai complexă de a se stabili planul de lucru săptămînal al maşinii, care furnizează beneficiul maxim, pentru diferite valori ale beneficiului unitar la reperul A.

Pentru rezolvare realizăm transformarea prealabilă a lui $\mathcal{P}(\lambda)$ în programul $P(\lambda)$: să se minimizeze

$$-f(x, y, \lambda) = -\lambda x - y, \tag{4.42}$$

cu restricţiile

$$3x + 2y + u = 3\,000, \quad x + v = 500, \quad y + w = 1\,000 \tag{4.43}$$

şi

$$x \geqslant 0, \quad y \geqslant 0, \quad u \geqslant 0, \quad v \geqslant 0, \quad w \geqslant 0. \tag{4.44}$$

Variabilele suplimentare introduse au semnificaţiile: $u/150$ = timpul de lucru săptămînal nefolosit al maşinii, v = capacitatea de stocaj nefolosită pentru reperul A, w = capacitatea de stocaj nefolosită pentru reperul B.

Grupul de paşi 1. Programul liniar parametric este sub o formă explicită primal admisibilă, aşa încît trecem la rezolvare folosind algoritmul simplex primal adaptat.

Scanned with OKEN Scanner

(A) Rezolvăm programul liniar ordinar $\pi_1(\lambda)$ obținut cînd considerăm că λ este o nedeterminată cu o valoare pozitivă oricît de mică în $P(\lambda)$:

	x	y	u	v	w	
$-f$	0	−1	0	0	0	0
	−1	0	0	0	0	0
u	3	2	1	0	0	3 000
v	1	0	0	1	0	500
w	0	①	0	0	1	1 000

⇓

	x	y	u	v	w	
$-f$	0	0	0	0	1	1 000
	−1	0	0	0	0	0
u	③	0	1	0	−2	1 000
v	1	0	0	1	0	500
y	0	1	0	0	1	1 000

⇓

	x	y	u	v	w	
$-f$	0	0	0	0	1	1 000
	0	0	1/3	0	−2/3	1 000/3
x	1	0	1/3	0	−2/3	1 000/3
v	0	0	−1/3	1	2/3	500/3
y	0	1	0	0	1	1 000

Problema $\pi_1(\lambda)$ este rezolvată: soluția de bază (S^1): $x = 1\,000/3$, $y = 1\,000$, $u = 0$, $v = 500/3$, $w = 0$, cu $f = 1\,000 + 1\,000\lambda/3$, este o soluție optimă pentru λ pozitiv oricît de mic.

Scanned with OKEN Scanner

(B) Condiţiile de optimalitate pentru această soluţie de bază sînt

$$\lambda/3 \geqslant 0, \quad 1 - \frac{2\lambda}{3} \geqslant 0, \tag{4.15}$$

şi sînt satisfăcute pentru $\lambda \in [0, 3/2]$; soluţia de bază (S^1) este o soluţie optimă a lui $P(\lambda)$ pe intervalul $\mathcal{J}_1 = [0, 3/2]$ şi nu este optimă pentru $\lambda > 3/2$.

(C) Realizăm schimbarea de parametru $\lambda = 3/2 + \nu$ şi obţinem un program liniar parametric $P_1(\nu)$, echivalent lui $P(\lambda)$, pe care trebuie să-l rezolvăm în grupul de paşi următor pentru $\nu \in [0, +\infty)$. Prima valoare critică a programului $P(\lambda)$ este $\lambda^1 = 3/2$.

Grupul de paşi 2. Programul liniar parametric $P_1(\nu)$ este sub formă explicită primal admisibilă.

(A) Rezolvăm programul liniar ordinar $\pi_2(\nu)$ obţinut cînd considerăm că ν este o nedeterminată cu o valoare pozitivă oricît de mică în $P_1(\nu)$. Schimbarea de parametru a fost executată astfel: s-a înmulţit în matricea formei finale a programului $P(\lambda)$, linia ce conţine coeficienţii lui λ din $-f(x, y, \lambda)$ cu valoarea critică $\lambda^1 = 3/2$ şi s-a adunat la linia termenilor liberi. Rezolvarea programului $\pi_2(\nu)$ se face într-un singur pas:

	x	y	u	v	w	
$-f$	0	0	1/2	0	0	1 500
	0	0	1/3	0	−2/3	1 000/3
x	1	0	1/3	0	−2/3	1 000/3
v	0	0	−1/3	1	(2/3)	500/3
y	0	1	0	0	1	1 000

⇓

	x	y	u	v	w	
$-f$	0	0	1/2	0	0	1 500
	0	0	0	1	0	500
x	1	0	0	1	0	500
w	0	0	−1/2	3/2	1	250
y	0	1	(1/2)	−3/2	0	750

Problema $\pi_2(\nu)$ este rezolvată: soluţia de bază (S^2): $x=500$, $y=750$, $u=0$, $v=0$, $w=250$ cu $f = 1\,500 + 500\nu = 750 + 500\lambda$, este o soluţie optimă pentru ν pozitiv oricît de mic.

Scanned with OKEN Scanner

(B) Codiţiile de optimalitate pentru această soluţie de bază se reduc la $\nu \geqslant 0$; soluţia de bază (S^2) este o soluţie optimă a lui $P_1(\nu)$ pe intervalul $[0, +\infty)$ şi deci a lui $P(\lambda)$ pe intervalul $[3/2, +\infty)$.

Rezolvarea programului $P(\lambda)$ este încheiată, iar tabelul soluţiilor optime pentru programul $\mathcal{P}(\lambda)$ este următorul:

	$\hat{x}$	$\hat{y}$	$\hat{f}$	interval
$(\hat{S}^1)$	1000/3	1000	$1000 + 1000\lambda/3$	$[0, 3/2]$
$(\hat{S}^2)$	500	750	$750 + 500\lambda$	$[3/2, +\infty)$

beneficiul maxim

$$f(\hat{x}, \hat{y}, \lambda) = \begin{cases} 1000 + \dfrac{1000}{3}\lambda, & \lambda \in [0, 3/2], \\ 750 + 500\lambda, & \lambda \in [3/2, +\infty), \end{cases} \tag{4.46}$$

este o funcţie liniară pe porţiuni şi poate fi folosită la rezolvarea unei serii de probleme. De exemplu, s-ar putea pune problema: pentru ce valori ale beneficiului unitar la reperul A procesul de producţie ar deveni rentabil în urma unei investiţii săptămînale de 2000 u.v., care ar permite creşterea acestui beneficiu unitar? Soluţionarea problemei revine la rezolvarea inecuaţiei $f(\hat{x}, \hat{y}, \lambda) - 2000 > 0$. Soluţia este $\lambda > 5/2$. De asemenea, cunoaşterea valorii critice $\lambda^1 = 3/2$ are importanţă practică, deoarece rata creşterii beneficiului unitar la reperul A se schimbă cînd λ trece de la valori mai mici la valori mai mari decît λ^1. Astfel, de exemplu, dacă prevăzînd un beneficiu unitar mai mic decît λ^1, punem în lucru un plan săptămînal optim dat de soluţia de bază (S^1) şi apoi, după realizarea producţiei, putem obţine un beneficiu unitar mai mare decît λ^1, atunci, soluţia pusă în lucru nefiind optimă, vom avea o pierdere de

$$f(\hat{x}_2, \hat{y}_2, \lambda) - f(\hat{x}_1, \hat{y}_1, \lambda) = \frac{500}{3}\lambda - 250. \tag{4.47}$$

4.4. PROGRAME LINIARE SUB FORMĂ EXPLICITĂ DUAL ADMISIBILĂ CU O NEDETERMINATĂ ÎN TERMENII LIBERI

Considerăm programul liniar $\rho(\nu)$: să se minimizeze

$$\chi(X, \nu) = \xi_0 + \nu\eta_0 + \sum_{k \in K} \zeta_0^k x_k, \tag{4.48}$$

cu restricţiile

$$x_i + \sum_{k \in K} \zeta_i^k x_k = \xi_i + \nu\eta_i, \ (i \in I), \tag{4.49}$$

şi

$$x_i \geqslant 0, \ x_k \geqslant 0, \ (i \in I, \ k \in K), \tag{4.50}$$

unde ν este un număr fixat pozitiv oricît de mic şi

$$\zeta_0^k \geqslant 0, \ (k \in K). \tag{4.51}$$

Deoarece $\rho(\nu)$ este un program liniar obişnuit de formă explicită dual admisibilă, pentru rezolvare vom aplica algoritmul simplex dual. Se ştie (v. §3.3) că operaţiile unui pas al algoritmului sînt: aplicarea criteriului de optimalitate, apoi dacă este necesară aplicarea criteriului de ieşire, urmează dacă este necesară aplicarea criteriului de intrare şi trecerea la o nouă formă explicită dual admisibilă. În continuare, ne interesează care sînt particularităţile acestor operaţii, datorite prezenţei numărului nedeterminat ν.

Să presupunem că au fost realizaţi r paşi ai algoritmului simplex dual, $(r \geqslant 0)$, pentru rezolvarea programului $\rho(\nu)$. Ca să nu mai complicăm notaţiile, vom admite că (4.48) — (4.50) este forma explicită dual admisibilă a lui $\rho(\nu)$, obţinută după cei r paşi şi vom trece la descrierea operaţiilor pasului $r+1$.

La aplicarea criteriului de optimalitate ne interesează să stabilim semnul unei expresii de forma $\xi_i + \nu\eta_i$, cînd ν are o valoare pozitivă oricît de mică. Semnul este dat de semnul lui ξ_i, dacă $\xi_i \neq 0$, şi de semnul lui η_i, dacă $\xi_i = 0$ şi $\eta_i \neq 0$. De aceea, criteriul de optimalitate

$$\xi_i + \nu\eta_i \geqslant 0, \ (i \in I), \tag{4.52}$$

se transcrie sub forma

$$\xi_i \geqslant 0, \ (i \in I),$$
$$\xi_i = 0 \Rightarrow \eta_i \geqslant 0. \tag{4.53}$$

Dacă am scrie termenii liberi în matricea cu care lucrăm la rezolvarea lui $\rho(\nu)$ pe două coloane în loc de una, folosind ultima coloană pentru termenii liberi din termenii liberi şi penultima

pentru coeficienţii lui ν, atunci forma (4.53) a criteriului se traduce astfel: elementele ultimei coloane trebuie să fie nenegative, iar elementele penultimei coloane corespunzătoare elementelor nule din ultima coloană trebuie să fie de asemenea nenegative. Desigur, criteriul (4.53) stabileşte dacă soluţia de bază

$$x_i = \xi_i + \nu\,\eta_i,\ x_k = 0,\ (i \in I,\quad k \in K), \tag{4.54}$$

este optimă, sau trebuie să trecem la operaţia următoare. În primul caz, obţinem şi valoarea funcţiei obiectiv $f = \xi_0 + \nu\eta_0$, care depinde de ν, ca şi soluţia de bază.

La aplicarea criteriului de ieşire ne interesează a compara expresiile $\xi_i + \nu\eta_i$, $(i \in I)$, pentru a găsi pe cea mai mică dintre ele. Este clar că dintre două expresii $\xi + \nu\eta$ şi $\xi' + \nu\eta'$ în care ν are o valoare pozitivă oricît de mică, aceea cu termenul liber mai mic va fi mai mică, iar dacă termenii liberi sînt egali, aceea cu coeficientul lui ν mai mic va fi mai mică. De aceea, criteriul de ieşire

$$\xi_j + \nu\eta_j = \min_i\,(\xi_i + \nu\eta_i), \tag{4.55}$$

se poate formula astfel: fie

$$\mathcal{J} = \{p \mid \xi_p = \min_i\,\xi_i\}; \tag{4.56}$$

dacă $|\mathcal{J}| = 1$, atunci x_j cu $j \in \mathcal{J}$ va deveni variabilă secundară, iar dacă $|\mathcal{J}| > 1$, atunci o variabilă x_j cu j determinat prin

$$\eta_j = \min_{p \in \mathcal{J}}\,\eta_p \tag{4.57}$$

va deveni secundară. Bineînţeles, ar fi posibil ca nici condiţia (4.57) să nu determine în mod unic variabila ce devine secundară, dar precizăm că în lipsa unicităţii se va alege o variabilă dintre cele ce satisfac (4.57). Dacă am întrebuinţa scrierea termenilor liberi pe două coloane în matricea formei explicite, criteriul de ieşire s-ar traduce astfel: se determină cel mai mic element al ultimei coloane, sau dacă există mai multe cu valoarea minimă, se determină cel mai mic element al penultimei coloane din limiile corespunzătoare elementelor cu vloarea minimă din ultima coloană. Desigur, după aplicarea criteriului de ieşire (4.56), (4.57) va trebui să cercetăm

Scanned with OKEN Scanner

elementele din linia variabilei x_j pentru a stabili fie că programul nu are soluţii admisibile, fie că este necesar să trecem la operaţia următoare.

La aplicarea criteriului de intrare, elementele utilizate sînt elemente din matricea sistemului şi coeficienţi din funcţia obiectiv şi cum aceştia nu depind de ν, criteriul de intrare va avea aspectul obişnuit şi va determina variabila x_h ce devine principală.

Pivotajul cu elementul pivot ζ_j^h, prin care se realizează trecerea la o nouă formă explicită dual admisibilă, se realizează de asemenea sub forma obişnuită.

Chiar şi fără aplicarea unei reguli de evitare a ciclării, în general, după un număr finit de paşi ai algoritmului simplex dual, programul $\rho(\upsilon)$ va fi rezolvat. Remarcăm că apariţia alternativelor ce decid oprirea algoritmului depinde valoarea lui ν.

Exemplul 4.4. Considerăm programul $\rho(\nu)$: să se minimizeze

$$\chi(X, \nu) = 2 + 3x_4 + x_5, \tag{4.58}$$

cu restricţiile

$$\left.\begin{aligned} x_1 \quad + x_4 + x_5 &= 3 - \nu \\ x_2 \quad + x_4 - x_5 &= -2 + \nu \\ x_3 + x_4 + 2x_5 &= 5 - 2\nu \end{aligned}\right\} \tag{4.59}$$

şi

$$x_1 \geqslant 0, \ldots, x_5 \geqslant 0, \tag{4.60}$$

unde ν este un număr pozitiv oricît de mic. Matricea asociată formei explicite dual admisibile disponibile este

$$\overline{\mathscr{A}} = \left(\begin{array}{ccccc|cc} 0 & 0 & 0 & 3 & 1 & 0 & -2 \\ \hline 1 & 0 & 0 & 1 & 1 & -1 & 3 \\ 0 & 1 & 0 & 1 & \textcircled{-1} & 1 & -2 \\ 0 & 0 & 1 & 1 & 2 & -2 & 5 \end{array}\right)$$

Elementul minim al ultimei coloane este negativ şi corespunde ecuaţiei cu variabila principală x_2, ce va deveni secundară; raportul maxim cerut de criteriul de intrare corespunde variabilei x_5, care devine principală. Pivotajul cu elementul pivot $\zeta_2^5 = -1$ conduce la matricea

$$\mathcal{A}^1 = \left(\begin{array}{ccccc|cc} 0 & 1 & 0 & 4 & 0 & 1 & -4 \\ \hline 1 & 1 & 0 & 2 & 0 & 0 & 1 \\ 0 & -1 & 0 & -1 & 1 & -1 & 2 \\ 0 & 2 & 1 & 3 & 0 & 0 & 1 \end{array}\right).$$

Elementul minim al ultimei coloane este pozitiv, soluția de bază $x_1 = 1, x_2 = 0, x_3 = 1, x_4 = 0, x_5 = 2 - \nu$ este o soluție optimă a programului $\rho(\nu)$, iar valoarea corespunzătoare a funcției obiectiv este $f = 4 - \nu$.

4.5. PROGRAME LINIARE PARAMETRICE CU PARAMETRUL ÎN TERMENII LIBERI

Considerăm un program liniar parametric $R(\lambda)$ de forma (b): să se minimizeze

$$f(X, \lambda) = c'X, \tag{4.61}$$

în condițiile

$$AX = b^0 + \lambda b^1, \; X \geqslant 0, \tag{4.62}$$

unde λ este un parametru cu valori într-un interval $\mathcal{J} = [0, \tau)$, τ fiind un număr pozitiv sau $+\infty$.

Presupunem că sistemul restricțiilor este explicitat; după cum se știe, această ipoteză este echivalentă compatibilității sistemului. Dacă sistemul restricțiilor n-ar fi explicitat, operația de explicitare, realizabilă de exemplu prin metoda eliminării complete, ar conduce fie la o formă explicită pentru orice valoare a lui λ din intervalul $\mathcal{J}$, fie la stabilirea faptului că sistemul este compatibil pentru o anumită valoare a lui λ din acest interval, fie la stabilirea incompatibilității sistemului pentru orice $\lambda \in \mathcal{J}$. Desigur, numai primul caz este de luat în considerare, deoarece în cazul al doilea programul liniar parametric este înlocuit printr-un program liniar ordinar, iar în cel de-al treilea caz problema rezolvării programului nu se mai pune.

Scanned with OKEN Scanner

Se înţelege că numai întîmplător prin explicitarea sistemului de restricţii se va obţine o formă explicită dual admisibilă a programului $R(\lambda)$. Se ştie însă (v. § 3.4.2) că pentru obţinerea unei forme explicite dual admisibile a lui $R(\lambda)$, după adăugarea unei restricţii artificiale conţinînd o nedeterminată, avem de rezolvat tot un program liniar parametric de tipul b), de formă explicită dual admisibilă. De aceea, să presupunem de la început că programul liniar parametric $R(\lambda)$ este de formă explicită dual admisibilă şi să trecem la descrierea rezolvării programului.

Rezolvarea programului liniar parametric $R(\lambda)$ se va realiza printr-un număr finit de grupuri de paşi ai algoritmului simplex dual, fiecare grup de paşi fiind aplicat unui program liniar parametric echivalent lui $R(\lambda)$ în baza unor schimbări de parametri şi a unor pivotaje succesive efectuate în grupurile de paşi anterioare. După cum se va arăta în cele ce urmează, orice grup de paşi rezolvă problema $R(\lambda)$ pe un subinterval al intervalului $(0, \tau)$.

Să admitem că s-au executat s grupuri de paşi, $s \geqslant 0$. Dacă $s > 0$, să presupunem că pe această cale s-a realizat rezolvarea lui $R(\lambda)$ pe intervalul $\mathcal{J}_s = [0, \lambda^s)$, unde $0 < \lambda^s < \tau$; să presupunem de asemenea că în baza schimbării de parametru $\lambda = \lambda^s + \nu$ şi a pivotajelor duale realizate în grupurile de paşi anterioare dispunem de un program liniar parametric $\rho(\nu)$ de formă explicită dual admisibilă. Atunci, înainte de a trece la grupul de paşi $s + 1$, dispunem oricare ar fi $s \geqslant 0$ de un program $\rho(\nu)$ de formă explicită dual admisibilă, astfel că $\rho(\lambda - \lambda^s)$ este echivalent cu $R(\lambda)$, $(\lambda^0 = 0)$. Programul $\rho(\nu)$ urmează a fi rezolvat pe intervalul $[0, \tau^s)$, $\tau^s = \tau - \lambda^s$, căci $R(\lambda)$ trebuie rezolvat încă pe intervalul $[\lambda^s, \tau)$. Să trecem acum la descrierea operaţiilor mari ale grupului de paşi $s + 1$.

(A) Se rezolvă programul liniar ordinar obţinut din programul liniar parametric $\rho(\nu)$ atunci cînd se consideră parametrul ν fixat la o valoare pozitivă oricît de mică. Rezolvarea se face prin algoritmul simplex dual, adaptat sub forma dată în paragraful precedent. Drept rezultat se obţine fie o soluţie de bază optimă (S^{s+1}) a acestui program liniar ordinar, fie concluzia că nu există soluţii admisibile.

(B) Se determină un interval $(0, \nu^*)$ cu proprietatea că pentru orice $\nu \in (0, \nu^*)$ programul $\rho(\nu)$ are aceeaşi rezolvare ca pentru ν pozitiv oricît de mic, cu alte cuvinte, pentru orice $\nu \in (0, \nu^*)$, fie (S^{s+1}) este o soluţie optimă a lui $\rho(\nu)$, fie acest program nu are soluţii

admisibile, după cum aceeaşi alternativă a fost stabilită cînd ν era un număr pozitiv fixat oricît de mic.

Acum, dacă $\nu^* \geqslant \tau^s$, programul liniar parametric $\rho(\nu)$ este rezolvat pe intervalul $(0, \tau^s)$; se va vedea că adesea dispunem şi de soluţia lui pentru $\nu = 0$. În acest caz, rezolvarea programului $R(\lambda)$ este încheiată. Dacă însă $\nu^* < \tau^s$, programul liniar parametric $\rho(\nu)$ este rezolvat pe intervalul $(0, \nu^*) \subset [0, \tau^s)$; se va vedea că dispunem şi de o soluţie optimă pentru $\nu = 0$. De aceea, în acest caz, rezolvarea programului $R(\lambda)$ este încheiată pe intervalul $\mathcal{J}_{s+1} = [0, \lambda^{s+1})$, $\lambda^{s+1} = \lambda^s + \nu^*$, dar trebuie să continue pe intervalul $[\lambda^{s+1}, \tau)$.

(C) Dacă $\nu^* < \tau^s$, se face în forma explicită dual admisibilă finală a lui $\rho(\nu)$ schimbarea de parametru $\nu = \nu^* + \theta$, unde θ este un nou parametru. Prin aceasta se obţine un program liniar parametric echivalent lui $R(\lambda)$ în baza schimbării de parametru $\lambda = \lambda^{s+1} + \theta$ şi a pivotajelor celor $s+1$ grupuri de paşi. Noul program parametric este sub o formă explicită dual admisibilă şi urmează a fi rezolvat pentru $\theta \in [0, \tau^{s+1})$, $\tau^{s+1} = \tau - \lambda^{s+1}$. Sîntem pregătiţi pentru a trece la grupul de paşi $s+2$.

Deşi procedeul de rezolvare este deja precizat, să trecem la analiza detaliată a operaţiilor (A), (B), (C), în vederea formulării exacte a rezultatelor grupului de paşi $s+1$.

(A) Să admitem că forma explicită dual admisibilă finală obţinută prin aplicarea operaţiei (A) asupra programului este următoarea: să se minimizeze

$$\chi(X, \nu) = \xi_0 + \nu\eta_0 + \sum_{k \in K} \zeta_0^k x_k, \tag{4.63}$$

cu restricţiile

$$x_i + \sum_{k \in K} \zeta_i^k x_k = \xi_i + \nu\eta_i, \ (i \in I), \tag{4.64}$$

şi

$$x_i \geqslant 0, \ x_k \geqslant 0, \ (i \in I, \ k \in K). \tag{4.65}$$

În (4.63) avem $\zeta_0^k \geqslant 0$, $(k \in K)$, şi întrucît (4.63) — (4.65) este forma explicită dual admisibilă finală obţinută după operaţia (A), am putut întîlni una dintre alternativele:

(A_1) $\xi_i + \nu\eta_i \geqslant 0$, $(i \in I)$, pentru ν pozitiv oricît de mic;

(A_2) $\xi_j + \nu\eta_j < 0$, pentru ν pozitiv oricît de mic şi un anumit $j \in I$, iar $\zeta_j^k \geqslant 0$, $(k \in K)$.

Scanned with OKEN Scanner

În cazul (A_1) soluția de bază

$$(S^{s+1}) \qquad x_i = \xi_i + \nu\eta_i,\ x_k = 0,\ (i \in I,\ k \in K), \tag{4.66}$$

este o soluție optimă a programului $\rho(\nu)$, iar în cazul (A_2) programul $\rho(\nu)$ nu are soluții admisibile, pentru ν pozitiv oricît de mic. Va trebui să ne ocupăm în continuare de ambele alternatve, pentru a stabili care poate fi rezultatul operației (B).

(B) Acum avem de determinat un interval cît mai mare pe care situația (A_1) sau (A_2), stabilită pentru o valoare pozitivă oricît de mică a lui ν, să se mențină.

(B_1) Dacă s-a obținut alternativa (A_1), înseamnă că sistemul de inecuații liniare

$$\xi_i + \nu\eta_i \geqslant 0,\ (i \in I), \tag{4.67}$$

care asigură optimalitatea soluției de bază (S^{s+1}) și care este compatibil pentru ν pozitiv oricît de mic, este format din inecuații de unul sau mai multe dintre următoarele trei tipuri:

$$\xi_i + \nu\eta_i \geqslant 0,\ (\xi_i > 0,\ \eta_i \geqslant 0); \tag{4.68}$$

$$\nu\eta_i \geqslant 0,\ (\eta_i \geqslant 0); \tag{4.69}$$

$$\xi_i + \nu\eta_i \geqslant 0,\ (\xi_i > 0,\ \eta_i < 0). \tag{4.70}$$

Definim ν^* prin

$$\nu^* = \begin{cases} \min\limits_{i \in I'} \left(-\dfrac{\xi_i}{\eta_i} \right), & I' \neq \emptyset, \\ +\infty, & I' = \emptyset, \end{cases} \tag{4.71}$$

unde

$$I' = \{i \mid i \in I,\ \xi_i > 0,\ \eta_i < 0\}. \tag{4.72}$$

Este clar că inecuațiile (4.67) sînt satisfăcute pentru orice $\nu \in [0, \nu^*)$ și dacă $\nu^* < +\infty$ cel puțin una dintre inecuații nu este satisfăcută pentru $\nu > \nu^*$; prin urmare, soluția de bază (S^{s+1}) este o soluție optimă a programului $\rho(\nu)$ pe intervalul $[0, \nu^*)$ și dacă $\nu^* < +\infty$ nu este optimă pe $(\nu^*, +\infty)$. Observăm că (S^{s+1}) este soluție optimă și pentru $\nu = \nu^*$ cînd $\nu^* < +\infty$. Programele $R(\lambda)$ și $\rho(\nu)$ sînt rezolvate pe intervalele corespunzătoare $[\lambda^s,\ \lambda^s + \nu^*)$ și $[0, \nu^*)$; desigur,

valorile optime corespunzătoare se exprimă diferit: $\xi_0 + (\lambda - \lambda^s)\eta_0$ și $\xi_0 + \nu\eta_0$.

(B_2) Dacă s-a obținut alternativa (A_2) înseamnă că inecuația

$$\xi_j + \nu\eta_j < 0, \tag{4.73}$$

este satisfăcută pentru ν pozitiv oricît de mic; prin urmare, inecuația (4.73) este de unul dintre următoarele trei tipuri:

$$\xi_j + \nu\eta_j < 0, \ (\xi_j < 0, \ \eta_j \leqslant 0); \tag{4.74}$$

$$\nu\eta_j < 0, \ (\eta_j < 0); \tag{4.75}$$

$$\xi_j + \nu\eta_j < 0, \ (\xi_j < 0, \ \eta_j > 0). \tag{4.76}$$

Definim ν^* prin

$$\nu^* = \begin{cases} -\dfrac{\xi_j}{\eta_j} & \text{cînd } \xi_j < 0, \quad \eta_j > 0, \\ \\ +\infty & \text{în celelalte cazuri.} \end{cases} \tag{4.77}$$

Este clar că inecuația (4.73) este satisfăcută pentru orice $\nu \in (0, \nu^*)$ și, dacă $\nu^* < +\infty$, nu este satisfăcută pentru $\nu \geqslant \nu^*$; prin urmare, programul $\rho(\nu)$ nu are soluții admisibile pe intervalul $(0, \nu^*)$. Observăm că pentru $\nu = 0$ programul ar putea avea soluții admisibile numai dacă inecuația (4.73) este de forma (4.75). Programele $R(\lambda)$ și $\rho(\nu)$ sînt rezolvate pe intervalele corespunzătoare $(\lambda^s, \lambda^s + \nu^*)$ și $(0, \nu^*)$; oricum intervalul pe care rezultatul obținut pentru λ pozitiv oricît de mic se menține este o semidreaptă, fie este vorba de semidreapta $(-\infty, \nu^*)$, fie de o semidreaptă care include $(0, +\infty)$.

Considerațiile precedente demonstrează teorema următoare:

Teorema 4.2. *Fiind dat un program liniar parametric $\rho(\nu)$ sub o formă explicită dual admisibilă, se poate determina ν^*, $\nu^* \in (0, +\infty)$, astfel că:*

— sau programul are o soluție optimă pe intervalul $[0, \nu^*)$, care pentru $\nu^* < +\infty$ este optimă chiar pe intervalul închis $[0, \nu^*]$ și nu este optimă pentru $\nu > \nu^*$;

— sau programul nu are soluții admisibile pe intervalul $(0, \nu^*)$, situație care se menține pentru $\nu = 0$ și nu are loc pentru $\nu > \nu^*$, cînd $\nu^* < +\infty$.

Scanned with OKEN Scanner

(C) Dacă $\nu^* \geqslant \tau^s$, programul parametric $R(\lambda)$ este rezolvat pe intervalul $[0, \tau)$, dacă însă $\nu^* < \tau^s$, este clar din teorema 4.2 că în ambele cazuri (A_1) şi (A_2) programul parametric $R(\lambda)$ este rezolvat pe intervalul $[0, \lambda^{s+1})$, $\lambda^{s+1} = \lambda^s + \nu^*$, ba chiar în primul caz şi pentru $\lambda = \lambda^{s+1}$. Schimbarea de parametru $\nu = \nu^* + \theta$ ne conduce la un program parametric sub o formă explicită dual admisibilă, echivalent lui $R(\lambda)$. Procedeul se reia cu un nou grup de paşi.

Din teorema 4.2, din faptul că numărul soluţiilor de bază ale sistemului (4.62) este finit şi din convergenţa algoritmului simplex dual rezultă convergenţa algoritmului simplex dual adaptat pentru rezolvarea programelor parametrice de tipul b). Situaţia finală obţinută în urma rezolvării programului $R(\lambda)$ prin $p + 1$ grupuri de paşi, $(p \geqslant 0)$, poate fi una dintre următoarele:

$p = 0$: pentru $\lambda \in (0, \tau)$ fie s-a găsit o soluţie optimă (S^1), fie s-a stabilit inexistenţa soluţiilor admisibile; în primul caz, (S^1) este soluţia optimă şi pentru $\lambda = 0$, în timp ce în cazul al doilea valoarea $\lambda = 0$ în general trebuie examinată separat;

$p > 0$: s-au determinat numerele $\lambda^1, \lambda^2, \ldots, \lambda^p$, $0 < \lambda^1 < \lambda^2 < \ldots < \lambda^p < \tau$, şi soluţiile de bază $(S^1), (S^2), \ldots, (S^p)$ încît (S^{s+1}) este o soluţie optimă a programului $R(\lambda)$ pentru orice $\lambda \in [\lambda^s, \lambda^{s+1}]$, $(s = 0, 1, \ldots, p-1)$, iar pentru $\lambda \in (\lambda^p, \tau)$ fie s-a găsit o soluţie de bază optimă (S^{p+1}), fie s-a stabilit inexistenţa soluţiilor admisibile.

Remarcăm că oricare dintre valorile λ^s care este extremitate comună la două intervale vecine, unde programul $R(\lambda)$ are soluţii optime, este o valoare a parametrului pentru care programul are măcar două soluţii de bază optime diferite, anume (S^s) şi (S^{s+1}); putem verifica imediat că în acest caz

$$X = \alpha X^s + (1 - \alpha) X^{s+1} \tag{4.78}$$

reprezintă de asemenea o soluţie optimă oricare ar fi α, $0 \leqslant \alpha \leqslant 1$, unde X^s şi X^{s+1} sînt cele două soluţii optime. Numerele $\lambda^1, \lambda^2, \ldots, \lambda^p$ se numesc *valori critice* ale programului liniar parametric $R(\lambda)$; cunoaşterea acestor numere au o importanţă practică deosebită.

Ne-am ocupat pînă acum de problema rezolvării unui program liniar parametric $R(\lambda)$ pe un interval de forma $[0, \tau)$, $\tau \in (0, +\infty)$.

Scanned with OKEN Scanner

Dacă intervalul pe care se cere a se rezolva programul $R(\lambda)$ este de forma $[\rho, \tau)$, unde ρ este un număr real oarecare, schimbarea de parametru $\lambda = \rho + \mu$ va transforma intervalul considerat în intervalul $[0, \tau - \rho)$. Dacă intervalul pe care se cere a se rezolva programul $R(\lambda)$ este de forma $(-\infty, +\infty)$, rezolvarea se poate face în două etape, mai întîi pe intervalul $[0, +\infty)$ şi apoi pe intervalul $(-\infty, 0)$ cu schimbarea de parametru prealabilă $\lambda = -\mu$.

O comparaţie a rezultatelor obţinute cu cele din § 4.3 relevă asemănări importante între algoritmul simplex primal adaptat pentru rezolvarea programelor parametrice de tip a) şi algoritmul simplex dual adaptat pentru rezolvarea programelor parametrice de tip b).

Exemplul 4.5. Considerăm programul liniar parametric $\mathcal{P}(\lambda)$: să se maximizeze

$$f = 2x + y, \tag{4.79}$$

cu restricţiile

$$x/50 + y/75 \leqslant 20 + \lambda, \; x \leqslant 500, \; y \leqslant 1000 \tag{4.80}$$

şi

$$x \geqslant 0, \; y \geqslant 0, \tag{4.81}$$

unde $\lambda \in [0, 20]$. Acesta este programul (4.5) — (4.7) din exemplul 4.1, program pe care ne propunem acum să-l rezolvăm. Semnificaţia parametrului λ este evidentă: timpul de lucru săptămînal suplimentar al maşinii. Vom cerceta în final chiar problema practică discutată la sfîrşitul exemplului 4.1.

Pentru rezolvare realizăm transformarea prealabilă a lui $\mathcal{P}(\lambda)$ în programul $R(\lambda)$: să se minimizeze

$$-f(x, y) = -2x - y, \tag{4.82}$$

cu restricţiile

$$\left.\begin{aligned} 3x + 2y + u &= 3\,000 + 150\,\lambda \\ x + v &= 500, \; y + w = 1\,000 \end{aligned}\right\} \tag{4.83}$$

şi

$$x \geqslant 0, \quad y \geqslant 0, \quad u \geqslant 0, \quad v \geqslant 0, \quad w \geqslant 0. \tag{4.84}$$

Variabilele suplimentare introduse au semnificaţiile din exemplul 4.3. Întrucît programul considerat nu este sub o formă explicită dual admisibilă, introducem ecuaţia artificială $x_0 + x + y = M$, (v. § 3.4.2), unde M este un număr pozitiv oricît de mare, şi condiţia $x_0 \geqslant 0$; printr-un pivotaj prealabil aducem problema lărgită la o formă explicită dual admisibilă şi apoi faza I continuă cu un pivotaj normal al metodei simplex duale, ţinînd seamă că M este un număr pozitiv oricît de mare:

Scanned with OKEN Scanner

	x	y	u	v	w	x_0		
$-f$	−2	−1	0	0	0	0	0	0
u	3	2	1	0	0	0	150	3 000
v	1	0	0	1	0	0	0	500
w	0	1	0	0	1	0	0	1 000
x_0	①	1	0	0	0	1	0	M

⇓

	x	y	u	v	w	x_0		
$-f$	0	1	0	0	0	2	0	$2M$
u	0	−1	1	0	0	(−3)	150	$3\,000-3M$
v	0	−1	0	1	0	−1	0	$500-M$
w	0	1	0	0	1	0	0	1 000
x	1	1	0	0	0	1	0	M

⇓

	x	y	u	v	w	x_0		
$-f$	0	1/3	2/3	0	0	0	100	2 000
x_0	0	1/3	−1/3	0	0	1	−50	$-1\,000+M$
v	0	−2/3	−1/3	1	0	0	−50	− 500
w	0	1	0	0	1	0	0	1 000
x	1	2/3	1/3	0	0	0	50	1 000

Scanned with OKEN Scanner

Cum variabila artificială x_0 a redevenit principală, a luat sfîrşit faza I, programul liniar parametric $R(\lambda)$ fiind adus la o formă explicită dual admisibilă; omitem coloana variabilei artificiale şi linia ecuaţiei artificiale şi trecem la faza II, rezolvarea problemei iniţiale.

Grupul de paşi 1. Programul liniar parametric este sub o formă explicită dual admisibilă, aşa încît trecem la rezolvare folosind algoritmul simplex dual adaptat.

(A) Rezolvăm programul liniar ordinar $\rho_1(\lambda)$ obţinut pentru λ cu o valoare pozitivă oricît de mică în $R(\lambda)$:

	x	y	u	v	w		
$-f$	0	1/3	2/3	0	0	100	2 000
v	0	(−2/3)	−1/3	1	0	−50	−500
w	0	1	0	0	1	0	1 000
x	1	2/3	1/3	0	0	50	1 000

⇓

	x	y	u	v	w		
$-f$	0	0	1/2	1/2	0	75	1 750
y	0	1	1/2	−3/2	0	75	750
w	0	0	−1/2	3/2	1	−75	250
x	1	0	0	1	0	0	500

Problema $\rho_1(\lambda)$ este rezolvată: soluţia de bază (S^1): $x = 500$, $y = 750 + 75\lambda$, $u = 0$, $v = 0$, $w = 250 - 75\lambda$, cu $f = 1750 + 75\lambda$ este optimă pentru λ pozitiv oricît de mic.

(B) Condiţiile de optimalitate pentru această soluţie de bază sînt

$$750 + 75\lambda \geqslant 0,\ 250 - 75\lambda \geqslant 0 \tag{4.85}$$

şi sînt satisfăcute pentru $\lambda \in [0,\ 10/3]$; soluţia de bază (S^1) este o soluţie optimă a lui $R(\lambda)$ pe intervalul $\mathcal{T}_1 = [0,\ 10/3]$ şi nu este optimă pentru $\lambda > 10/3$.

Scanned with OKEN Scanner

(C) Realizăm schimbarea de parametru $\lambda = 10/3 + \nu$ și obținem un program liniar parametric $R_1(\nu)$, echivalent lui $R(\lambda)$, pe care trebuie să-l rezolvăm în grupul de pași următor pentru $\nu \in [0, 50/3]$. Prima valoare critică a programului $R(\lambda)$ este $\lambda^1 = 10/3$.

Grupul de pași 2. Programul liniar parametric $R_1(\nu)$ este sub o formă explicită dual admisibilă.

(A) Rezolvăm programul liniar ordinar $\rho_2(\nu)$ obținut cînd considerăm că ν este o nedeterminată cu o valoare pozitivă oricît de mică în $R_1(\nu)$. Schimbarea de parametru a fost executată astfel: s-a înmulțit în matricea formei finale a programului $R(\lambda)$, coloana ce conține coeficienții lui λ din termenii liberi cu valoarea critică $\lambda^1 = 10/3$ și s-a adunat la coloana termenilor liberi. Rezolvarea programului $\rho_2(\nu)$ se face într-un singur pas:

	x	y	u	v	w		
$-f$	0	0	1/2	1/2	0	75	2 000
y	0	1	1/2	−3/2	0	75	1 000
w	0	0	(−1/2)	3/2	1	−75	0
x	1	0	0	1	0	0	500

⇓

	x	y	u	v	w		
$-f$	0	0	0	2	1	0	2 000
y	0	1	0	0	1	0	1 000
u	0	0	1	−3	−2	150	0
x	1	0	0	1	0	0	500

Problema $\rho_2(\nu)$ este rezolvată: soluția de bază (S^2): $x = 500$, $y = 1000$, $u = 150\nu$, $v = 0$, $w = 0$, cu $f = 2000$ este optimă pentru ν pozitiv oricît de mic.

(B) Condițiile de optimalitate pentru această soluție de bază se reduc la $\nu \geqslant 0$; soluția de bază (S^2) este o soluție optimă a lui $R_1(\nu)$ pe intervalul $(0, +\infty)$ și deci,

Scanned with OKEN Scanner

cu transformarea $\nu = \lambda - 10/3$, a lui $R(\lambda)$ pe intervalul [10/3, 20]. Rezolvarea programului $R(\lambda)$ este încheiată, iar tabloul soluţiilor optime pentru programul $\mathcal{R}(\lambda)$ este următorul:

	$\hat{x}$	$\hat{y}$	$\hat{f}$	Interval
$\hat{S}^1$	500	$750+75\lambda$	$1750+75\lambda$	[0, 10/3]
$\hat{S}^2$	500	1 000	2 000	[10/3, 20]

Funcţia beneficiul maxim

$$f(\hat{x}, \hat{y}, \lambda) = \begin{cases} 1750 + 75\lambda, & \lambda \in [0, 10/3], \\ 2\,000, & \lambda \in [10/3, 20], \end{cases} \tag{4.86}$$

este o funcţie liniară pe porţiuni şi poate fi folosită la rezolvarea problemelor puse în exemplul 4.1. Prima concluzie este că o prelungire a timpului de lucru al maşinii realizată la un cost pe oră prelungită de k unităţi valorice va fi rentabilă numai dacă $k < 75$, altminteri beneficiul suplimentar

$$\beta(\hat{x}, \hat{y}, \lambda) = f(\hat{x}, \hat{y}, \lambda) - f(\hat{x}, \hat{y}, 0) - k\lambda, \tag{4.87}$$

dat de

$$\beta(\hat{x}, \hat{y}, \lambda) = \begin{cases} (75 - k)\,\lambda, & \lambda \in [0, 10/3], \\ 250 - k\lambda, & \lambda \in [10/3, 20], \end{cases} \tag{4.88}$$

este negativ sau nul. A doua concluzie este că sporirea timpului de lucru al maşinii, chiar la un cost orar de $k < 75$ unităţi valorice, este inutilă dincolo de limita $\lambda^* = 20 + 10/3$, cum o arată funcţia (4.88). Se observă că sporirea suplimentară a beneficiului este împiedicată pentru $\lambda > \lambda^*$ de insuficienţa capacităţilor de stocaj.

Scanned with OKEN Scanner

5

TEORIA DUALITĂȚII PROGRAMELOR LINIARE

5.1. PROBLEME DUALE SIMETRICE

Fie o problemă de programare liniară (P): să se minimizeze

$$f = \sum_{k=1}^{n} c_k x_k, \tag{5.1}$$

cu restricțiile

$$\sum_{k=1}^{n} a_i^k x_k \geqslant b_i, \ (i = 1, \ldots, n), \tag{5.2}$$

și

$$x_k \geqslant 0, \ (k = 1, \ldots, n). \tag{5.3}$$

Acestei probleme i se poate asocia o altă problemă de programare liniară (D): să se maximizeze

$$g = \sum_{i=1}^{m} b_i w_i, \tag{5.4}$$

cu restricțiile

$$\sum_{i=1}^{m} a_i^k w_i \leqslant c_k, \ (k = 1, \ldots, n), \tag{5.5}$$

$$w_i \geqslant 0, \ (i = 1, \ldots, m). \tag{5.6}$$

Datorită proprietăților de legătură dintre cele două programe liniare se spune că problema (D) este *duală* problemei (P); cînd se discută despre perechea de probleme (P) și (D), problema (P) este denumită problema primală iar problema (D) este denumită problema duală. Dacă se scrie problema (D) ca problemă de minimizare și se înmulțesc toate restricțiile sale cu -1 pentru inversarea sensurilor inecuațiilor, apoi se construiește problema asociată, la fel cum s-a dedus (D) din (P), se vede imediat că se obține problema (P); cu alte cuvinte, duala dualei este problema primală.

Scanned with OKEN Scanner

Astfel, poziția de primală sau dual convine oricăreia dintre cele două probleme; în continuare vom atribui problemei (P) poziția de primală și lui (D) poziția de dual.

Cel mai simplu construim problema dual a unei probleme date cu ajutorul diagramei următoare:

	$x_1 \dots x_k \dots x_n$	g	
w_1	$a_1^1 \dots a_1^k \dots a_1^n$	$\geqslant$	b_1
$\vdots$	$\vdots \quad \vdots \quad \vdots$	$\vdots$	$\vdots$
w_i	$a_i^1 \dots a_i^k \dots a_i^n$	$\geqslant$	b_i
$\vdots$	$\vdots \quad \vdots \quad \vdots$	$\vdots$	$\vdots$
w_m	$a_m^1 \dots a_m^k \dots a_m^n$	$\geqslant$	b_m
f	$\leqslant \dots \leqslant \dots \leqslant$	max / min	
	$c_1 \dots c_k \dots c_n$		

Este clar că liniile diagramei pot fi completate imediat ce se dă problema (P); atunci problema (D) se poate citi cu ușurință pe coloane.

Discernem următoarele legături de construcție între cele două probleme: a) pentru fiecare inecuație a problemei primale (duale) corespunde cîte o variabilă a problemei duale (primale); b) coeficienții variabilelor dintr-o inecuație a primalei (dualei) sînt coeficienții variabilei corespunzătoare în inecuațiile dualei (primalei); c) termenii liberi ai problemei primale (duale) devin coeficienții variabilelor în funcția obiectiv a problemei duale (primale). În afară de aceasta, semnele $\geqslant$ din primală se înlocuiesc în dual prin semnele $\leqslant$, iar sensul optimizării se inversează.

Pentru stabilirea proprietăților principale ale perechilor de probleme duale folosim următoarea scriere matriceală a problemelor:

$$(P) \qquad \min\{c'X \mid AX \geqslant b,\ X \geqslant 0\}$$

și

$$(D) \qquad \max\{W'b \mid W'A \leqslant c',\ W \geqslant 0\},$$

unde toți vectorii folosiți sînt vectori coloană.

Teorema 5.1. *Dacă $\bar{X}$ și $\bar{W}$ sînt soluții admisibile ale problemelor (P) și (D) respectiv, iar $\bar{f}$ și $\bar{g}$ valorile corespunzătoare ale funcțiilor obiectiv, atunci $\bar{f} \geqslant \bar{g}$.*

Scanned with OKEN Scanner

Demonstrație. Folosim identitatea

$$\bar{f} - \bar{g} = (c' - \bar{W}'A)\bar{X} + \bar{W}'(A\bar{X} - b). \tag{5.7}$$

Deoarece $\bar{X}$ și $\bar{W}$ sînt soluții admisibile ale celor două probleme, avem

$$\begin{aligned} c' - \bar{W}'A \geqslant 0, \qquad & \bar{X} \geqslant 0, \\ \bar{W}' \geqslant 0, \qquad & A\bar{X} - b \geqslant 0, \end{aligned} \tag{5.8}$$

așa încît din (5.7) rezultă $\bar{f} \geqslant \bar{g}$.

Se știe că dacă un program liniar are soluții admisibile, atunci el poate avea fie soluție optimă, fie o funcție obiectiv care nu-i mărginită pe mulțimea soluțiilor admisibile (v. § 3.4.1). Atunci, din teorema 5.1 deducem imediat teorema următoare:

Teorema 5.2. *Dacă una dintre problemele* (P) *sau* (D) *are funcția obiectiv nemărginită, inferior pentru problema* (P) *sau superior pentru problema* (D), *atunci cealaltă problemă nu are soluții admisibile.*

Demonstrație. Să presupunem că problema (P) are funcția obiectiv nemărginită inferior pe mulțimea soluțiilor admisibile. Să admitem că problema (D) ar avea o soluție admisibilă $\bar{W}$ și valoarea corespunzătoare a funcției obiectiv este $\bar{g}$. Atunci, cum funcția obiectiv f a problemei (P) nu este mărginită inferior, ar exista o soluție admisibilă $\bar{X}$ pentru care $\bar{f} < \bar{g}$; dar, ultima inegalitate este imposibilă, căci contrazice teorema 5.1. O demonstrație analoagă se poate face pentru afirmația corespunzătoare cazului nemărginirii superioare a funcției g.

O altă consecință a teoremei 5.1 este teorema următoare:

Teorema 5.3. *Dacă* $\bar{X}$ *și* $\bar{W}$ *sînt soluții admisibile ale problemelor* (P) *și* (D) *respectiv, iar valorile corespunzătoare* $\bar{f}$ *și* $\bar{g}$ *ale funcțiilor obiectiv sînt egale, atunci* $\bar{X}$ *și* $\bar{W}$ *sînt soluții optime.*

Demonstrație. Dacă $\bar{X}$ n-ar fi soluție optimă a problemei (P), înseamnă că ar exista o soluție admisibilă X^* astfel ca $f^* < \bar{f}$. Cum $\bar{f} = \bar{g}$, am deduce că perechea de soluții admisibile X^* și $\bar{W}$ dă valori f^* și $\bar{g}$ ale funcțiilor obiectiv astfel ca $f^* < \bar{g}$, ceea ce ar contrazice teorema 5.1. La fel stabilim optimalitatea soluției $\bar{W}$.

Să arătăm că totdeauna valorile optime ale unei perechi de probleme duale sînt egale și că existența unei soluții optime pentru una dintre probleme atrage existența unei soluții optime pentru cealaltă. Pentru aceasta stabilim în prealabil teorema următoare:

Scanned with OKEN Scanner

Teorema 5.4. *Dacă dispunem de o formă explicită a probemei*

$$(P') \qquad \min\{c'X \mid AX - Y = b,\ X \geqslant 0,\ Y \geqslant 0\},$$

atunci vectorul coeficienților funcției obiectiv din forma explicită reprezintă o soluție de bază a problemei

$$(D') \qquad \max\{W'b \mid W'A + U' = c',\ W \geqslant 0,\ U \geqslant 0\};$$

dacă soluția de bază corespunzătoare formei explicite a problemei (P') este optimă, atunci și soluția de bază a problemei (D') este optimă, iar valorile funcțiilor obiectiv sînt egale.

Demonstrație. Una dintre formele explicite ale problemei (P') este chiar

$$\min\{c'X \mid -AX + Y = -b,\ X \geqslant 0,\ Y \geqslant 0\}, \tag{5.9}$$

adică aceea în care variabilele suplimentare sînt toate principale. Soluția de bază corespunzătoare formei explicite este $\bar{X} = 0$, $\bar{Y} = -b$, iar $\bar{f} = 0$ și vectorul coeficienților funcției obiectiv este $(c'\,O')'$. Punem $\bar{U} = c$, $\bar{W} = 0$ și observăm că vectorul $(\bar{U}'\,\bar{W}')'$ reprezintă o soluție de bază a problemei (D') cu $\bar{g} = 0$. Dacă soluția de bază a problemei (P') este optimă, adică $b \leqslant 0$ și $c \geqslant 0$, observăm că și soluția de bază a problemei (D') este optimă; avem totodată $\bar{f} = \bar{g}$. Să considerăm acum și alte forme explicite ale problemei (P'). Putem avea două situații: a) variabilele suplimentare sînt toate secundare; b) unele variabile suplimentare sînt principale.

a) Să admitem că $x_1, \ldots, x_m$ sînt variabilele principale, ceea ce nu restrînge generalitatea. Desigur, o condiție necesară pentru a întîlni această situație este $m \leqslant n$. Să notăm $X^1 = (x_1, \ldots, x_m)'$ și $X^2 = (x_{m+1}, \ldots, x_n)'$ și să împărțim matricea A în mod corespunzător, $A = (A^1A^2)$. Cum A^1 este matricea coeficienților variabilelor principale, avem $|A^1| \neq 0$. Sistemul restricțiilor se scrie

$$A^1X^1 + A^2X^2 - Y = b, \tag{5.10}$$

astfel încît forma sa explicită care ne interesează devine

$$X^1 + (A^1)^{-1}A^2X^2 - (A^1)^{-1}Y = (A^1)^{-1}b. \tag{5.11}$$

Împărțind vectorul c în mod corespunzător și folosind sistemul (5.11) pentru eliminarea variabilelor principale, găsim următoarea

Scanned with OKEN Scanner

expresie a funcției obiectiv:

$$f=(c^1)'\cdot(A^1)^{-1}b+[(c^2)'-(c^1)'\cdot(A^1)^{-1}A^2]X^2+ \\ +(c^1)'\cdot(A^1)^{-1}Y. \tag{5.12}$$

Soluția de bază corespunzătoare formei explicite este

$$\bar{X}^1=(A^1)^{-1}b,\ \bar{X}^2=0,\ \bar{Y}=0,\ \text{cu}\ \bar{f}=(c^1)'\cdot(A^1)^{-1}b; \tag{5.13}$$

vectorul coeficienților funcției obiectiv (5.12) este

$$(O'\,|\,(c^2)'-(c^1)'\cdot(A^1)^{-1}A^2\,|\,(c^1)'\cdot(A^1)^{-1})'. \tag{5.14}$$

Dacă punem

$$\bar{U}=(O'\,|\,(c^2)'-(c^1)'\cdot(A^1)^{-1}A^2)',\ \bar{W}=((c^1)'\cdot(A^1)^{-1})', \tag{5.15}$$

putem verifica imediat că $\bar{W}'A+\bar{U}'=c'$ și valoarea funcției obiectiv pentru această soluție de bază a problemei (D') este $\bar{g}=(c^1)'\cdot(A^1)^{-1}b$. Dacă (5.13) este o soluție de bază optimă a problemei (P'), atunci vectorul (5.14) este nenegativ, deci (5.15) este o soluție de bază admisibilă a problemei (D') și deoarece $\bar{g}=\bar{f}$, rezultă din teorema 5.3 că soluția de bază a lui (D') este optimă.

b) Să presupunem că $x_1,\ldots,x_p,\ y_{p+1},\ldots,y_m$ sînt variabilele principale ale formei explicite, ceea ce nu restrînge generalitatea. Matricea sistemului $AX-Y=b$ se poate scrie sub forma

$$\begin{pmatrix} A_1^1 & A_1^2 & -E_p & O^2 \\ A_2^1 & A_2^2 & O^1 & -E_{m-p} \end{pmatrix},$$

unde A_1^1, A_1^2, A_2^1, A_2^2, O^1, O^2 sînt matrice de dimensiunile $p\times p$, $p\times(n-p)$, $(m-p)\times p$, $(m-p)\times(n-p)$, $(m-p)\times p$, $p\times(m-p)$ respectiv. Acestei împărțiri în blocuri a matricei sistemului îi corespunde o împărțire evidentă a vectorului termenilor liberi și a vectorilor X și Y. Astfel, sistemul se transcrie

$$\begin{aligned} A_1^1X^1+A_1^2X^2-Y^1 &= b^1, \\ A_2^1X^1+A_2^2X^2-Y^2 &= b^2, \end{aligned} \tag{5.16}$$

iar forma sa explicită ce ne interesează este

$$\begin{aligned} &X^1+(A_1^1)^{-1}A_1^2X^2-(A_1^1)^{-1}Y^1=(A_1^1)^{-1}b^1, \\ &-[A_2^2-A_2^1(A_1^1)^{-1}A_1^2]X^2-A_2^1(A_1^1)^{-1}Y^1+Y^2=-[b^2-A_2^1(A_1^1)^{-1}b^1]. \end{aligned} \tag{5.17}$$

Scanned with OKEN Scanner

Împărțind vectorul c în mod corespunzător și folosind numai primele p ecuații din sistemul (5.17) pentru eliminarea variabilelor principale, găsim următoarea expresie a funcției obiectiv:

$$f = (c^1)' \cdot (A_1^1)^{-1} b^1 + [(c^2)' - (c^1)' \cdot (A_1^1)^{-1} A_1^2] X^2 + (c^1)' \cdot (A_1^1)^{-1} Y^1. \quad (5.18)$$

Soluția de bază corespunzătoare formei explicite este

$$\bar{X}^1 = (A_1^1)^{-1} b^1, \ \bar{X}^2 = 0, \ \bar{Y}^1 = 0, \ \bar{Y}^2 = -[b^2 - A_2^1 (A_1^1)^{-1} b^1], \quad (5.19)$$

cu $\bar{f} = (c^1)' \cdot (A_1^1)^{-1} b^1$; vectorul coeficienților funcției obiectiv este

$$(O' \vdots (c^2)' - (c^1)' \cdot (A_1^1)^{-1} A_1^2 \vdots (c^1)' \cdot (A_1^1)^{-1} \vdots O')'. \quad (5.20)$$

Dacă punem

$$\bar{U} = (O' \vdots (c^2)' - (c^1)' \cdot (A_1^1)^{-1} A_1^2)', \ \bar{W} = ((c^1)' \cdot (A_1^1)^{-1} \vdots O'), \quad (5.21)$$

putem verifica imediat că $\bar{W}'A + \bar{U}' = c'$ și $\bar{g} = (c^1)'(A_1^1)^{-1} b^1$ este valoarea funcției obiectiv pentru această soluție de bază a problemei (D'). Dacă (5.19) este o soluție de bază optimă a problemei (P'), atunci vectorul (5.20) este nenegativ, deci (5.21) este o soluție de bază admisibilă a problemei (D') și cum $\bar{f} = \bar{g}$, rezultă din teorema 5.3 că soluția de bază a lui (D') este optimă.

O consecință imediată a teoremei 5.4 este teorema următoare:

Teorema 5.5. *Dacă una dintre problemele (P) sau (D) are soluție optimă, atunci cealaltă problemă are de asemenea soluție optimă, iar valorile corespunzătoare ale funcțiilor obiectiv sînt egale.*

După cum se vede, teorema 5.5 afirmă nu numai că reciproca teoremei 5.3 este adevărată, ci și faptul că în formularea unei asemenea reciproce s-ar presupune în mod inutil existența soluțiilor optime la ambele probleme (P) și (D). Din teorema 5.4 rezultă și următorul fapt important din punct de vedere practic: în matricea finală a formei explicite a problemei (P') rezolvate prin metoda simplex primală sau duală se găsește atît o soluție optimă a problemei (P'), cît și o soluție optimă a problemei (D'); rezultă că am rezolvat în același timp ambele probleme. Prin aceasta, am rezolvat atît problema (P), cît și problema (D); soluțiile lor optime sînt, respectiv, vectorul $\bar{X}$ din soluția optimă $(\bar{X}' \ \bar{Y}')'$ a problemei (P') și vectorul $\bar{W}$ din soluția optimă $(\bar{U}' \bar{W}')'$ a problemei (D'). Mai precis, se observă că $\bar{W}$ reprezintă vectorul coeficienților variabilelor $y_1, \ldots, y_m$ din forma finală a funcției obiectiv.

Avînd în vedere cele cinci teoreme demonstrate, există doar următoarele patru posibilități logice de rezolvabilitate a unei perechi

Scanned with OKEN Scanner

de probleme duale (P) şi (D):

1) (P) şi (D) nu au soluţii admisibile;

2) (P) nu are soluţii admisibile şi (D) are funcţia obiectiv nemărginită superior;

3) (P) are funcţia obiectiv nemărginită inferior şi (D) nu are soluţii admisibile;

4) (P) şi (D) au soluţii optime şi valorile funcţiilor obiectiv sînt egale.

Alternativa 1) este posibilă nu numai logic; se poate construi un exemplu în care să întîlnim exact această situaţie.

Teorema 5.6. *Soluţiile optime ale perechii de probleme duale* (P) şi (D) *sînt soluţiile sistemului de inecuaţii şi ecuaţii*

$$AX \geqslant b, \quad X \geqslant 0, \tag{5.22}$$

$$W'A \leqslant c', \quad W \geqslant 0, \tag{5.23}$$

$$(c' - W'A)X = 0, \quad W'(AX - b) = 0. \tag{5.24}$$

Demonstraţie. Dacă $\bar{X}$, $\bar{W}$ este o pereche de soluţii optime ale problemelor (P) şi (D) respectiv, atunci evident $\bar{X}$ satisface (5.22), $\bar{W}$ satisface (5.23), iar din identitatea (5.7), în care primul membru este nul conform teoremei 5.5 iar termenii din membrul drept sînt nenegativi, deducem că împreună $\bar{X}$ şi $\bar{W}$ satisfac (5.24). Reciproc, dacă $\bar{X}$ şi $\bar{W}$ satisfac (5.22), (5.23), (5.24), există o pereche de soluţii admisibile pentru care $\bar{f} = \bar{g}$, şi din teorema 5.3 deducem că vectorii reprezintă o pereche de soluţii optime.

Pentru o pereche de vectori X, W vom denumi: (5.22) — condiţii de admisibilitate primală, (5.23) — condiţii de admisibilitate duală, iar (5.24) — condiţii de complementarietate.

Exemplul 5.1. Considerăm problema (P): să se minimizeze

$$f = 2x_1 - x_2, \tag{5.25}$$

cu restricţiile

$$-x_1 - x_2 \geqslant -3, \quad 3x_1 - x_2 \geqslant -1 \tag{5.26}$$

şi

$$x_1 \geqslant 0, \; x_2 \geqslant 0. \tag{5.27}$$

Problema duală lui (P) este problema (D): să se maximizeze

$$g = -3w_1 - w_2, \tag{5.28}$$

cu restricţiile

$$-w_1 + 3w_2 \leqslant 2, \quad -w_1 - w_2 \leqslant -1 \tag{5.29}$$

şi

$$w_1 \geqslant 0, \; w_2 \geqslant 0. \tag{5.30}$$

Scanned with OKEN Scanner

Vom găsi o pereche de soluții optime pentru problemele (P) și (D) respectiv, rezolvînd numai una dintre ele, anume problema (P).

După introducerea variabilelor suplimentare, condițiile (5.26), (5.27) devin

$$x_1 + x_2 + y_1 = 3, \qquad -3x_1 + x_2 + y_2 = 1, \tag{5.31}$$

$$x_1 \geqslant 0,\ x_2 \geqslant 0,\ y_1 \geqslant 0,\ y_2 \geqslant 0,$$

așa încît dispunem de o formă explicită primal admisibilă a problemei (P'); trecem la aplicarea algoritmului simplex primal:

	x_1	x_2	y_1	y_2	
f	2	−1	0	0	0
y_1	1	1	1	0	3
y_2	−3	①	0	1	1

⇓

	x_1	x_2	y_1	y_2	
f	−1	0	0	1	1
y_1	④	0	1	−1	2
x_2	−3	1	0	1	1

⇓

	x_1	x_2	y_1	y_2	
f	0	0	1/4	3/4	3/2
x_1	1	0	1/4	−1/4	1/2
x_2	0	1	3/4	1/4	5/2

Scanned with OKEN Scanner

Din ultimul tabel obținem o soluție optimă pentru problema (P) $\bar{x}_1 = 1/2$, $\bar{x}_2 = 5/2$, și o soluție optimă pentru problema (D) : $\bar{w}_1 = 1/4$, $\bar{w}_2 = 3/4$; valoarea comună a funcțiilor obiectiv este $\bar{f} = \bar{g} = -3/2$. Se înțelege, prima soluție optimă a fost extrasă din soluția optimă a problemei (P') dată de tablou, iar a doua soluție optimă a fost extrasă din vectorul coeficienților funcției obiectiv.

5.2. PROBLEME DUALE NESIMETRICE

Fie o problemă de programare liniară (π): să se minimizeze

$$f = \sum_{k=1}^{n} c_k x_k, \tag{5.32}$$

cu restricțiile

$$\sum_{k=1}^{n} a_i^k x_k = b_i, \quad (i = 1, \ldots, m) \tag{5.33}$$

și

$$x_k \geqslant 0, \quad (k = 1, \ldots, n). \tag{5.34}$$

Deși programul (π) nu este de forma (5.1), (5.2), (5.3), putem să-i asociem un program dual. Pentru aceasta, înlocuim sistemul restricțiilor (5.33) prin sistemul echivalent de inecuații

$$\begin{aligned} &\sum_{k=1}^{n} a_i^k x_k \geqslant b_i, \qquad (i = 1, \ldots, m), \\ &-\sum_{k=1}^{n} a_i^k x_k \geqslant -b_i, \qquad (i = 1, \ldots, m). \end{aligned} \tag{5.35}$$

Acum se poate construi duala problemei (5.32), (5.35), (5.34) după procedeul din paragraful precedent. Dacă se notează variabilele duale asociate inecuațiilor (5.35) cu w_i^1 și w_i^2, $(i = 1, \ldots, m)$, programul dual lui (π) se scrie : să se maximizeze

$$g = \sum_{i=1}^{m} b_i (w_i^1 - w_i^2), \tag{5.36}$$

cu restricțiile

$$\sum_{i=1}^{m} a_i^k (w_i^1 - w_i^2) \leqslant c_k, \qquad (k = 1, \ldots, n), \tag{5.37}$$

și

$$w_i^1 \geqslant 0, \; w_i^2 \geqslant 0, \qquad (i = 1, \ldots, m). \tag{5.38}$$

Este clar că putem introduce variabilele $w_i = w_i^1 - w_i^2$, $(i = 1, \ldots, m)$, care nu vor fi supuse la condițiile de nenegativitate, așa încît programul dual lui (π) este problema (Δ): să se maximizeze

$$g = \sum_{i=1}^{m} b_i w_i, \tag{5.39}$$

cu restricțiile

$$\sum_{i=1}^{m} a_i^k w_i \leqslant c_k, \qquad (k = 1, \ldots, n). \tag{5.40}$$

Perechea de probleme (π), (Δ) nu mai are aspectul simetric al perechii de probleme (P), (D); de aceea, perechea de probleme (P), (D) este denumită *pereche de probleme simetrice*, în timp ce perechea de probleme (π), (Δ) este denumită *pereche de probleme nesimetrice*. Proprietățile stabilite pentru perechile de probleme simetrice se mențin și pentru cele nesimetrice. Teoremele 5.1, 5.2, 5.3, 5.5, care nu conțin în enunțuri elemente date ale perechii de probleme duale, se mențin ad litteram pentru perechile de probleme nesimetrice. Ne vom opri asupra teoremei analoage teoremei 5.4, întrucît pe această cale vom arăta cum se rezolvă simultan o pereche de probleme nesimetrice.

Fie $\bar{X}$ o soluție de bază optimă a programului (π), iar X_p și X_s vectori formați cu variabilele principale și respectiv secundare ale soluției de bază. Notăm cu A matricea sistemului restricțiilor (5.33), $A = (a_i^k)$; fie A_p și A_s submatricele lui A ce conțin coloanele corespunzătoare variabilelor din X_p și X_s respectiv. Notăm cu c vectorul coeficienților funcției obiectiv (5.32), $c = (c_k)$; fie c_p și c_s vectorii ce conțin coeficienții variabilelor din X_p și X_s respectiv. În sfîrșit, notăm cu b vectorul termenilor liberi din (5.33), $b = (b_i)$. Atunci sistemul (5.33) se scrie sub forma

$$A_p X_p + A_s X_s = b, \tag{5.41}$$

iar sub forma explicită corespunzătoare soluției de bază optime

$$X_p + A_p^{-1} A_s X_s = A_p^{-1} b. \tag{5.42}$$

Funcția obiectiv exprimată cu ajutorul variabilelor secundare se scrie

$$f = c_p' A_p^{-1} b + (c_s' - c_p' A_p^{-1} A_s) X_s. \tag{5.43}$$

Scanned with OKEN Scanner

Soluția de bază optimă a programului (π) este

$$\bar{X}_p = A_p^{-1}b, \qquad \bar{X}_s = 0, \tag{5.44}$$

iar valoarea optimă

$$\bar{f} = c_p' A_p^{-1}b. \tag{5.45}$$

Condițiile de admisibilitate și optimalitate pentru soluția de bază $\bar{X}$ sînt

$$A_p^{-1}b \geqslant 0, \qquad c_s' - c_p'A_p^{-1}A_s \geqslant 0. \tag{5.46}$$

Arătăm că vectorul

$$\bar{W}' = c_p'A_p^{-1}, \tag{5.47}$$

reprezintă o soluție optimă a programului (Δ). Avem

$$c' - \bar{W}'A = (c_p', c_s') - c_p'A_p^{-1}(A_p, A_s), \tag{5.48}$$

unde am scris atît vectorul c cît și matricea A cu elementele așezate în blocuri; rezultă

$$c' - \bar{W}'A = (O', c_s' - c_p'A_p^{-1}A_s), \tag{5.49}$$

și, ținînd seamă de (5.46), deducem $c' - \bar{W}'A \geqslant 0$, adică $\bar{W}$ este o soluție admisibilă a programului (Δ). Valoarea corespunzătoare a funcției obiectiv este

$$\bar{g} = c_p'A_p^{-1}b, \tag{5.50}$$

și, ținînd seamă de (5.45), deducem $\bar{f} = \bar{g}$. Atunci, conform teoremei 5.3, vectorul $\bar{W}$ dat prin (5.47) este o soluție optimă a programului (Δ). Cu aceasta am demonstrat teorema următoarea:

Teorema 5.4'. *Dacă $\bar{X}$ este o soluție de bază optimă a programului (π), și A_p este submatricea coloanelor coeficienților din A corespunzătoare variabilelor principale, iar c_p vectorul coeficienților acelorași variabile din funcția obiectiv, atunci $\bar{W}' = c_p'A_p^{-1}$ reprezintă o soluție optimă a programului (Δ) și $\bar{f} = \bar{g}$.*

Scanned with OKEN Scanner

Remarcăm că vectorul $\overline{W}$ nu apare ca un vector al coeficienților funcției obiectiv, scrise sub forma (5.43).

Pentru a demonstra o teoremă analoagă teoremei 5.6 ne vom baza pe identitatea

$$f - g = (c' - W'A)\,X + W'(AX - b); \tag{5.51}$$

deoarece pentru orice pereche de soluții admisibile ale celor două probleme al doilea termen din (5.51) este nul și pentru optimalitatea lor este necesară și suficientă egalitatea valorilor funcțiilor obiectiv, rezultă că am stabilit teorema următoare:

Teorema 5.6'. *Soluțiile optime ale perechii de probleme* (π) *și* (Δ) *sînt soluțiile sistemului de inecuații și ecuații*

$$\left.\begin{aligned} &AX = b, \quad X \geqslant 0 \\ &W'A \leqslant c' \\ &(c' - W'A)\,X = 0 \end{aligned}\right\} \tag{5.52}$$

Exemplul 5.2. Să rezolvăm problema (Δ): să se maximizeze

$$g = w_1 + 2w_2, \tag{5.53}$$

cu restricțiile

$$-w_1 + 2w_2 \leqslant 4, \quad w_1 - w_2 \leqslant 4. \tag{5.54}$$

Problema este de forma (5.39), (5.40), întrucît ea este duală problemei (π): să se minimizeze

$$f = 4x_1 + 4x_2, \tag{5.55}$$

cu restricțiile

$$\begin{aligned} &-x_1 + x_2 = 1, \quad 2x_1 - x_2 = 2, \\ &x_1 \geqslant 0, \quad x_2 \geqslant 0. \end{aligned} \tag{5.56}$$

Problema (π) are soluția admisibilă unică $\bar{x}_1 = 3$, $\bar{x}_2 = 4$, care este deci și soluția optimă. Variabilele fiind ambele principale, avem

$$A_p = \begin{pmatrix} -1 & 1 \\ 2 & -1 \end{pmatrix}, \qquad A_p^{-1} = \begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix}, \tag{5.57}$$

Cum $c_p' = (4\ 4)$, din (5.47) găsim o soluție optimă a problemei (Δ), anume

$$\overline{W}' = (4 \quad 4)\begin{pmatrix} 1 & 1 \\ 2 & 1 \end{pmatrix}, \tag{5.58}$$

adică $\bar{w}_1 = 12$, $\bar{w}_2 = 8$. Se verifică imediat egalitatea $\bar{f} = \bar{g} = 28$.

5.3. DIAGRAMA LUI A. W. TUCKER

Rezultatele date în § 5.1 și § 5.2 se pot da pentru probleme aparent mai generale decît problemele (5.1) — (5.3) și (5.32) — (5.34). Cu alte cuvinte, pentru noi tipuri de probleme vom putea construi de asemenea problema duală asociată și vor avea loc teoreme foarte asemănătoare teoremelor 5.1 — 5.6.

Să considerăm problema (π^*) : să se minimizeze

$$\sum_{k=1}^{n} \zeta_k x_k, \tag{5.59}$$

cu restricțiile

$$\begin{aligned} &\sum_{k=1}^{n} \xi_{i_1}^k x_k \geqslant \eta_{i_1}, \quad (i_1 = 1, \ldots, m_1), \\ &\sum_{k=1}^{n} \xi_{i_2}^k x_k = \eta_{i_2} \quad (i_2 = m_1 + 1, \ldots, m), \end{aligned} \tag{5.60}$$

și

$$\begin{aligned} &x_{k_1} \geqslant 0, \quad (k_1 = 1, \ldots, n_1), \\ &x_{k_2} \text{ oarecare}, \ (k_2 = n_1 + 1, \ldots, n). \end{aligned} \tag{5.61}$$

Remarcăm, în afară de prezența simultană a ecuațiilor și inecuațiilor în sistemul (5.60), și faptul că problema poate avea variabile neconstrînse la condiții de nenegativitate.

Problemei (π^*) îi putem asocia o problemă care să merite denumirea de problemă duală, deoarece putem să o transformăm în prealabil astfel ca să devină de unul din tipurile (5.1) — (5.3) sau (5.32) —(5.34). Într-adevăr, pe lîngă transformarea ecuațiilor în inecuații, sau a inecuațiilor în ecuații, deja cunoscute, mai putem realiza și schimbarea variabilelor neconstrînse în variabile constrînse, astfel : oricare variabilă neconstrînsă x poate fi înlocuită cu perechea de variabile

Scanned with OKEN Scanner

constrînse x^+ și x^- legate de x prin $x = x^+ - x^-$. Dacă am proceda în acest mod, vom observa cu ușurință că obținem următoarea problemă (Δ^*) : să se maximizeze

$$\sum_{i=1}^{m} \eta_i w_i, \tag{5.62}$$

cu restricțiile

$$\begin{aligned} &\sum_{i=1}^{m} \xi_i^{k_1} w_i \leqslant \zeta_{k_1}, \quad (k_1 = 1, \ldots, n_1), \\ &\sum_{i=1}^{m} \xi_i^{k_2} w_i = \zeta_{k_2}, \quad (k_2 = n_1 + 1, \ldots, n), \end{aligned} \tag{5.63}$$

și

$$\begin{aligned} &w_{i_1} \geqslant 0, \quad (i_1 = 1, \ldots, m_1), \\ &w_{i_2} \text{ oarecare}, \quad (i_2 = m_1 + 1, \ldots, m). \end{aligned} \tag{5.64}$$

Deoarece problema (Δ^*) este construită din (π^*) la fel ca (D) din (P), teoremele 5.1, 5.2, 5.3, 5.5 se mențin ad litteram, iar teoremele 5.4, 5.6 sînt înlocuite prin enunțuri asemănătoare.

Vom da numai formularea teoremei analoage teoremei 5.6.

Teorema 5.6.* *Soluțiile optime ale perechii de probleme* (π^*) *și* (Δ^*) *sînt soluțiile sistemului*

$$\begin{aligned} &\xi_1 X \geqslant \eta_1, \qquad \xi_2 X = \eta_2, \quad X_1 \geqslant 0, \\ &W' \xi^1 \leqslant \zeta_1', \qquad W' \xi^2 = \zeta_2', \quad W_1 \geqslant 0, \\ &(\zeta_1' - W' \xi^1) X_1 = 0, \; W_1' (\xi_1 X - \eta_1) = 0, \end{aligned} \tag{5.65}$$

unde elementele problemei (π^*) *sînt*

$$\begin{aligned} &\xi_1 = (\xi_1^1 \;\; \xi_2^1), \\ &\xi_2 = (\xi_2^1 \;\; \xi_2^2), \end{aligned} \qquad \xi^1 = \begin{pmatrix} \xi_1^1 \\ \xi_2^1 \end{pmatrix}, \qquad \xi^2 = \begin{pmatrix} \xi_1^2 \\ \xi_2^2 \end{pmatrix},$$

$$\xi = \begin{pmatrix} \xi_1^1 & \xi_1^2 \\ \xi_2^1 & \xi_2^2 \end{pmatrix}, \qquad \eta = \begin{pmatrix} \eta_1 \\ \eta_2 \end{pmatrix}, \qquad \zeta = \begin{pmatrix} \zeta_1 \\ \zeta_2 \end{pmatrix}, \tag{5.66}$$

iar vectorii variabilelor sînt împărțiți corespunzător.

Scanned with OKEN Scanner

Desigur, calea cea mai simplă de a construi problema duală (Δ^*), cînd problema primală (π^*) este dată, rămîne aşa-numita diagramă a lui A. W. Tucker:

(Δ^*) / (π^*)	$\dots\ x_{k_1}\ \dots\ x_{k_2}\ \dots$	g^*	
	$\dots \geqslant 0\ \dots\ -\ \dots$		
w_{i_1}	$\dots\ \xi_{i_1}^{k_1}\ \dots\ \xi_{i_1}^{k_2}\ \dots$	$\geqslant$	η_{i_1}
w_{i_2}	$\dots\ \xi_{i_2}^{k_1}\ \dots\ \xi_{i_2}^{k_2}\ \dots$	$=$	η_{i_2}
f^*	$\dots \leqslant \dots = \dots$	max	
	$\dots\ \zeta_{k_1}\ \dots\ \zeta_{k_2}\ \dots$	min	

Ca şi în cazul construcţiei problemei duale pentru problema (5.1) — (5.3), după ce s-au completat liniile cu datele problemei (π^*) se poate citi problema (Δ^*) pe coloane.

Menţionăm că teoremele 5.6, 5.6′, 5.6*, care dau condiţii necesare şi suficiente de extrem legat, sînt cazuri particulare ale teoremei lui H. Kuhn şi A. Tucker, care are un rol central în programarea neliniară.

Deosebit de interesantă şi importantă pentru teoria modelelor macroeconomice este interpretarea economică a elementelor perechilor de probleme duale. Această interpretare conduce la concluzii practice importante în politica economică.

Exemplul 5.3. Fie problema (π^*): să se minimizeze

$$f^* = -x_1 + 2x_2 - x_3, \tag{5.67}$$

cu restricţiile

$$\left.\begin{aligned} x_1 + x_2 + x_3 &\geqslant -2 \\ -x_1 - 2x_2 + x_3 &= 1 \\ -x_2 &\geqslant -3 \end{aligned}\right\} \tag{5.68}$$

și

$$x_3 \geqslant 0. \tag{5.69}$$

Ne propunem de a construi diagrama lui A. W. Tucker. Pentru aceasta completăm liniile cu datele problemei:

(Δ^*) / (π^*)	x_1	x_2	x_3	g^*	
	$-$	$-$	$\geqslant$		
w_1	1	1	1	$\geqslant$	-2
w_2	-1	-2	1	$=$	1
w_3	0	-1	0	$\geqslant$	-3
f^*				max	
	-1	2	-1	min	

apoi completăm spațiul rămas, pe baza regulii că variabilă neconstrînsă corespunde unei ecuații și reciproc. Obținem de pe coloane problema (Δ^*): să se maximizeze

$$g^* = -2w_1 + w_2 - 3w_3, \tag{5.70}$$

cu restricțiile

$$\left.\begin{aligned} w_1 - w_2 &= -1 \\ w_1 - 2w_2 - w_3 &= 2 \\ w_1 + w_2 &\leqslant -1 \end{aligned}\right\} \tag{5.71}$$

și

$$w_1 \geqslant 0, \qquad w_3 \geqslant 0. \tag{5.72}$$

Uneori, problema duală se rezolvă simplu și permite rezolvarea rapidă a problemei primale. Așa s-a întîmplat în exemplul 5.2 și se întîmplă și aici. Eliminînd w_2 din primele două condiții (5.71), găsim condiția $-w_1 - w_3 = 4$ și cum $w_1 = -w_3 - 4 \geqslant 0$, rezultă $w_3 \leqslant -4$ în contradicție cu $w_3 \geqslant 0$. Așadar, problema (Δ^*) nu are soluții admisibile. Problema (π^*) are soluții admisibile; de exemplu, una este $x_1 = 1$, $x_2 = -\frac{1}{2}$, $x_3 = 1$. Din teoremele de dualitate rezultă că problema (π^*) are o funcție obiectiv nemărginită inferior. Această concluzie se verifică și direct: dînd lui x_2, de exemplu, valoarea zero, iar lui x_1 și x_3 orice valori pozitive astfel ca $x_3 = x_1 + 1$, sînt verificate condițiile (5.68), (5.69), prin urmare, crescînd oricît de mult și simultan valorile variabilelor x_1 și x_3, dar în așa fel ca $x_3 = x_1 + 1$, valoarea funcției obiectiv va scădea oricît de mult. Remarcăm, că pentru probleme de forma (5.59) — —(5.61) se înțelege prin soluție admisibilă un vector X care verifică (5.60), (5.61).

5.4. LEGĂTURA DINTRE ALGORITMUL SIMPLEX PRIMAL ŞI ALGORITMUL SIMPLEX DUAL

Considerăm perechea de probleme duale simetrice

$$(P) \qquad \min\{c'X \mid AX \geqslant b, \quad X \geqslant 0\},$$

şi

$$(D) \qquad \max\{W'b \mid W'A \leqslant c', \quad W \geqslant 0\}.$$

După introducerea în cele două probleme (P) şi (D) a unor vectori Y şi U conţinînd variabile suplimentare, făcută în vederea rezolvării, se obţine perechea de probleme

$$(P') \qquad \min\{c'X \mid -AX + Y = -b, \quad X \geqslant 0, \ Y \geqslant 0\},$$

$$(D') \qquad \min\{W'(-b) \mid W'A + U' = c', \quad W \geqslant 0, \ U \geqslant 0\}.$$

Ceea ce ne va preocupa în continuare va fi să stabilim ce legătură există între algoritmul simplex primal aplicat problemei (D') şi algoritmul simplex dual aplicat problemei (P').

Teorema 5.6 arată că soluţiile optime ale perechii de probleme (P) şi (D) sînt soluţiile sistemului de condiţii

$$-AX + Y = -b, \ X \geqslant 0, \ Y \geqslant 0, \tag{5.73}$$

$$W'A + U' = c', \ U \geqslant 0, \ W \geqslant 0, \tag{5.74}$$

$$U'X = 0, \ W'Y = 0. \tag{5.75}$$

Condiţiile (5.75), ţinînd seamă de condiţiile de nenegativitate din (5.73), (5.74), pot fi înlocuite prin următoarele:

$$u_k x_k = 0, \quad w_i y_i = 0, \quad (i = 1, \ldots, m; \quad k = 1, \ldots, n). \tag{5.76}$$

Cu alte cuvinte, perechile de soluţii optime ale problemelor (P) şi (D) sînt perechi de soluţii admisibile ale problemelor (P') şi (D') care satisfac condiţiile (5.76). Aceste condiţii arată că variabilele

Scanned with OKEN Scanner

duale se grupează în perechi duale și în orice pereche de soluții optime cel puțin cîte o variabilă din fiecare pereche dual are valoarea zero.

Vom numi soluții *complementare* ale problemelor (P') și (D') oricare două soluții care satisfac condițiile (5.76). Atunci este clar că orice pereche de soluții complementare care mai sînt și admisibile este o pereche de soluții optime. Se înțelege, există și perechi de soluții de bază complementare, dintre care cele admisibile sînt soluții de bază optime. De exemplu, $\bar{X} = 0$, $\bar{Y} = -b$ și $\bar{U} = c$, $\bar{W} = 0$ reprezintă o pereche de soluții de bază complementare. Ținînd seamă de identitatea (5.7), care a folosit de fapt la demonstrarea teoremei 5.6, observăm că valorile funcțiilor obiectiv ale problemelor (P) și (D) sînt egale pentru orice pereche de soluții complementare, iar ale problemelor (P') și (D') sînt egale în valoare anbolută și de semn contrar. De exemplu, pentru perechea de soluții de bază complementare menționată mai sus, avem $\bar{f} = \bar{g} = 0$, unde $\bar{f} = c'\bar{X}$, $\bar{g} = \bar{W}'b$.

După familiarizarea cu acest nou concept, de pereche de soluții complementare, să trecem la studiul legăturii dintre cei doi algoritmi simpliciali.

Observăm că sistemele restricțiilor din problemele (P') și (D') sînt sub o formă explicită, iar variabilele principale nu apar în funcțiile obiectiv respective. Problema (D') a fost luată ca problemă de minimizare, deoarece în capitolele precedente s-a lucrat exclusiv cu problemele de minimizare și va fi astfel mai ușor de urmărit textul în continuare.

Soluțiile de bază corespunzătoare celor două forme explicite sînt respectiv

$$\bar{X} = 0, \quad \bar{Y} = -b \tag{5.77}$$

și

$$\bar{U} = c, \quad \bar{W} = 0. \tag{5.78}$$

Din (5.77), (5.78) și relațiile $f = c'X$, $-g = W(-b)$ rezultă că soluția (5.77) este admisibilă, dacă și numai dacă soluția (5.78) este dual admisibilă și soluția (5.77) este dual admisibilă, dacă și numai dacă soluția (5.78) este admisibilă. De aceea, condiția de optimalitate a perechii de soluții de bază complementare (5.77), (5.78) este fie admisibilitatea ambelor soluții, fie admisibilitatea dual a ambelor soluții, fie admisibilitatea primală și dual a uneia dintre soluții.

Scanned with OKEN Scanner

Să presupunem că problema (D') este sub o formă explicită primal admisibilă, adică $c \geqslant 0$; totodată problema (P') este sub o formă explicită dual admisibilă. Să aplicăm acum algoritmul simplex primal problemei (D') și algoritmul simplex dual problemei (P') și să comparăm operațiile unui pas în ambele cazuri.

1) Criteriul de optimalitate:

$-b \geqslant 0$ $\Rightarrow$ (5.77) este soluție optimă a lui (D') conform criteriului algoritmului primal:
$\Rightarrow$ (5.78) este soluție optimă a lui (P') conform criteriului algoritmului dual.

2) Dacă $\exists i$, $-b_i < 0$, atunci se trece în ambele cazuri la determinarea unui indice j prin condiția $-b_j = \min_i (-b_i)$, și avem

$-b_j = \min_i (-b_i)$ $\Rightarrow$ variabila w_j devine principală conform criteriului de intrare al algoritmului primal;
$\Rightarrow$ variabila y_j devine secundară conform criteriului de ieșire al algoritmului dual.

Acum se poate întîmpla ca

$a_j^k \geqslant 0$, $(\forall k)$ $\Rightarrow$ funcția obiectiv $-g$ a lui (D') este nemărginită superior;
$\Rightarrow$ nu există soluții admisibile pentru (P').

3) Dacă $\exists k$, $a_j^k < 0$, atunci se trece la determinarea unui indice h, prin condiția

$$\frac{c_h}{a_j^h} = \min_{k | a_j^k < 0} \left(\frac{c_k}{a_j^k} \right), \tag{5.79}$$

care se mai poate scrie și sub forma

$$-\frac{c_h}{a_j^h} = \max_{k | -a_j^k > 0} \left(-\frac{c_k}{a_j^k} \right); \tag{5.80}$$

de aici

(5.79) $\Rightarrow$ variabila u_h devine secundară conform criteriului de ieșire al algoritmului primal;
(5.80) $\Rightarrow$ variabila x_h devine principală conform criteriului de intrare al algoritmului dual.

4) În ambele cazuri se realizează un pivotaj cu elementul pivot a_j^k, anume: un pivotaj primal pentru problema (D'), respectiv un pivotaj dual pentru problema (P'). În urma acestor pivotaje se

obțin două forme explicite ale acestor probleme, una primal admisibilă pentru problema (D') și una dual admisibilă pentru problema (P'). Soluțiile de bază corespunzătoare formează tot o pereche de soluții de bază complementare, căci după pivotaj perechile de variabile duale (x_h, u_h) și (y_j, w_j) ce conțin măcar cîte o variabilă nulă în (5.77), (5.78), devenită nenulă în noile soluții de bază, vor conține și acum măcar cîte o variabilă nulă: u_h și y_j. Astfel și valorile funcțiilor obiectiv pentru noile soluții de bază, deși s-au modificat, vor rămîne în continuare opuse. Se mai poate arăta că între noile forme explicite, să zicem (P'') și (D''), există aceeași legătură ca între (P') și (D'), cu alte cuvinte, (D'') se obține din duala lui (P'') prin adăugarea variabilelor principale prezente în (D'') ca variabile suplimentare și reciproc.

În acest mod, am reușit să arătăm că algoritmul simplex dual pentru o problemă (P') aflată sub o formă explicită dual admisibilă reprezintă de fapt algoritmul simplex primal pentru problema asociată (D') aflată sub o formă explicită primal admisibilă.

Remarcăm faptul că proprietatea de complementarietate a unor perechi de soluții ale celor două probleme (P') și (D') este un caz particular de ortogonalitate, anume cînd doi vectori ortogonali au proprietatea suplimentară de a avea măcar o componentă nulă în orice pereche de componente cu aceeași poziție.

Exemplul 5.4. Considerăm problema (P): să se minimizeze

$$f = -5 + x_1 + 2x_2 + 3x_3, \tag{5.81}$$

cu restricțiile

$$x_1 - x_2 + 2x_3 \geqslant 2, \quad -x_1 + x_2 + x_3 \geqslant 3 \tag{5.82}$$

și

$$x_1 \geqslant 0, \quad x_2 \geqslant 0, \quad x_3 \geqslant 0. \tag{5.83}$$

Problema duală este problema (D): să se maximizeze

$$g = -5 + 2w_1 + 3w_2, \tag{5.84}$$

cu restricțiile

$$w_1 - w_2 \leqslant 1, \quad -w_1 + w_2 \leqslant 2, \quad 2w_1 + w_2 \leqslant 3, \tag{5.85}$$

și

$$w_1 \geqslant 0, \quad w_2 \geqslant 0. \tag{5.86}$$

Dacă introducem variabile suplimentare în sistemul (5.82) și înmulțim cu -1 ambele ecuații obținute, găsim problema (P'): să se minimizeze

$$f = -5 + x_1 + 2x_2 + 3x_3, \tag{5.87}$$

Scanned with OKEN Scanner

cu restricţiile

$$-x_1 + x_2 - 2x_3 + y_1 = -2,\ x_1 - x_2 - x_3 + y_2 = -3 \tag{5.88}$$

şi

$$x_1 \geqslant 0,\ x_2 \geqslant 0,\ x_3 \geqslant 0,\ y_1 \geqslant 0,\ y_2 \geqslant 0. \tag{5.89}$$

Dacă introducem variabile suplimentare în sistemul (5.85) şi în loc de a maximiza funcţia g vom minimiza funcţia $h = -g$, găsim problema (D') : să se minimizeze

$$h = 5 - 2w_1 - 3w_2, \tag{5.90}$$

cu restricţiile

$$u_1 + w_1 - w_2 = 1,\ u_2 - w_1 + w_2 = 2,\ u_3 + 2w_1 + w_2 = 3 \tag{5.91}$$

şi

$$u_1 \geqslant 0,\ u_2 \geqslant 0,\ u_3 \geqslant 0,\ w_1 \geqslant 0,\ w_2 \geqslant 0. \tag{5.92}$$

Recunoaştem problema (P') ca fiind problema (2.63) — (2.65), căreia i s-a aplicat o succesiune de pivotaje duale în exemplul 2.5; de asemenea, recunoaştem problema (D') ca fiind problema (2.51) — (2.53), căreia i s-a aplicat o succesiune de pivotaje primale în exemplul 2.4. Dealtfel, exemplele 2.4 şi 2.5 au fost alese tocmai cu scopul ca acum să putem compara pivotajele realizate în paşii corespunzători ai celor doi algoritmi simpleciali şi să ilustrăm legătura lor prezentată teoretic mai înainte. Perechile de soluţii de bază complementare obţinute succesiv sînt : perechea iniţială

$$(0,\ 0,\ 0,\ -2,\ -3), \qquad (1,\ 2,\ 3,\ 0,\ 0),$$

perechea determinată după primul pas

$$(0,\ 3,\ 0,\ -5,\ 0) \qquad (3,\ 0,\ 1,\ 0,\ 2),$$

perechea optimă determinată după al doilea pas

$$(0,\ 4/3,\ 5/3,\ 0,\ 0), \qquad (3,\ 0,\ 0,\ 1/3,\ 7/3).$$

Remarcăm în matricele formelor explicite succesive ale problemei (P') prezenţa în linia funcţiei obiectiv a soluţiei de bază corespunzătoare a problemei (D') şi în matricele formelor explicite succesive ale probemei (D') prezenţa în linia funcţiei obiectiv a soluţiei de bază corespunzătoare a problemei (P'). Dacă se suprimă în formele explicite succesive ale celor două probleme variabilele principale şi se înlocuiesc în restricţii semnele de egalitate cu semnul $\leqslant$, se constată că după fiecare pas concomitent al celor doi algoritmi, perechea de probleme obţinută este o pereche de probleme duale simetrice.

Scanned with OKEN Scanner

6

PROGRAME LINIARE ÎN NUMERE ÎNTREGI

6.1. O ALTĂ FORMĂ DE PIVOTAJ

În metodele de rezolvare a programelor în numere întregi, bazate pe algoritmii simpleciali pentru rezolvarea programelor liniare obișnuite, nu se lucrează cu acea matrice lărgită care a fost asociată programului liniar în capitolele precedente. Așa cum se va vedea mai departe, cauza este că în cursul rezolvării se adaugă noi restricții și această operație de adăugare se realizează mult mai bine dacă se folosește o altă matrice asociată unui program liniar. De aceea, înainte de a se trece la considerarea programelor liniare în numere întregi se va defini o nouă matrice asociată unui program liniar obișnuit și se va arăta cum funcționează algoritmii cunoscuți asupra acestei matrice.

Fie programul liniar sub o formă explicită (P) : să se minimizeze

$$f = f_0 + \sum_{k \in K} \alpha_0^k x_k, \tag{6.1}$$

cu restricțiile

$$x_i + \sum_{k \in K} \alpha_i^k x_k = \beta_i, \quad (i \in I), \tag{6.2}$$

și

$$x_i \geqslant 0, \; x_k \geqslant 0, \; (i \in I, \; k \in K), \tag{6.3}$$

unde I și K reprezintă ca de obicei mulțimile de indici ale variabilelor principale și secundare, respectiv.

Asupra programului (P) se vor realiza în prealabil unele transformări : a) egalitatea (6.1) se va scrie sub forma

$$x_0 + \sum_{k \in K} \alpha_0^k x_k = \beta_0, \tag{6.4}$$

unde $x_0 = -f$ reprezintă o nouă variabilă, desigur principală și neconstrînsă, iar $\beta_0 = -f_0$, și se va adăuga sistemului (6.2); b) egalitățile

$$x_k + (-x_k) = 0, \quad (k \in K), \tag{6.5}$$

Scanned with OKEN Scanner

vor fi adăugate de asemenea sistemului (6.2). Cu toate că (6.5) sînt niște identități vom întrebuința pentru acestea denumirea de ecuații banale întîlnită în literatură.

După transformările a), b) putem scrie împreună ecuațiile (6.2), (6.4), (6.5) sub forma

$$x_i + \sum_{k \in K} \alpha_i^k x_k = \beta_i, \ (i = 0, 1, \ldots, n). \tag{6.6}$$

Evident, problema a fost numai formal modificată; dealtfel, ecuația (6.6) obținută pentru $i = 0$ o vom denumi ecuația zero și își va păstra rolul din algoritmii simpliciali. Remarcăm că sistemul (6.6) cuprinde exprimarea tuturor variabilelor problemei (P) ca funcții de variabilele secundare ale formei explicite.

Vom nota: $\bar{C} = (\alpha_i^k)$, $\bar{d} = (\beta_i)$, $\bar{\mathscr{C}} = (\bar{C}, d)$, încît $\bar{C}$, $\bar{d}$, $\bar{\mathscr{C}}$ vor fi matrice de dimensiunile $(n+1) \times (n-m)$, $(n+1) \times 1$, $(n+1) \times (n-m+1)$, dacă aveam m variabile principale și $n-m$ variabile secundare în forma explicită (6.1) — (6.3). Matricea $\bar{\mathscr{C}}$ este asociată în mod unic formei explicite și, reciproc, forma explicită este unic determinată prin această matrice; astfel, putem lucra cu matricea $\bar{\mathscr{C}}$ la trecerea de la programul considerat la unul echivalent, reconstituind forma explicită sub care a ajuns programul după un număr de pivotaje, din ultima matrice obținută.

Dacă se compară matricea $\bar{\mathscr{C}}$ tocmai definită cu matricea $\bar{\mathscr{A}}$, asociată formei explicite (6.1) — (6.3), se constată că din $\bar{\mathscr{A}}$ se obține $\bar{\mathscr{C}}$ astfel: se suprimă coloanele corespunzătoare variabilelor principale, se rearanjează liniile corespunzătoare variabilelor principale în ordinea indicilor lor și se introduc în poziții convenabile linii banale corespunzătoare variabilelor secundare; o linie banală are elementele nule cu excepția unuia egal cu -1 într-o anumită coloană.

Exemplul 6.1. Să reconsiderăm exemplul 3.2 și să construim matricea $\bar{\mathscr{C}}$, asociată formei explicite (3.21) — (3.23). Scriem ecuația corespunzătoare funcției (3.21) sub forma

$$x_0 - 2x_1 - 3x_3 = -5; \tag{6.7}$$

adăugăm această ecuație, împreună cu ecuațiile banale

$$x_1 - x_1 = 0, \quad x_3 - x_3 = 0, \tag{6.8}$$

la sistemul (3.22) și rearanjînd ecuațiile din noul sistem în ordinea din formula (6.6), obținem

$$\left.\begin{aligned} x_0 - 2x_1 - 3x_3 &= -5 \\ x_1 - x_1 &= 0 \\ x_2 - x_1 + x_3 &= 1 \\ x_3 - x_3 &= 0 \\ x_4 + 2x_1 - x_3 &= 2 \\ x_5 - x_1 + x_3 &= 3 \end{aligned}\right\} \qquad (6.9)$$

și matricea

$$\bar{\mathscr{C}} = \left(\begin{array}{cc|c} -2 & -3 & -5 \\ \hline -1 & 0 & 0 \\ -1 & 1 & 1 \\ 0 & -1 & 0 \\ 2 & -1 & 2 \\ -1 & 1 & 3 \end{array}\right).$$

Transformările de la matricea $\bar{\mathscr{A}}$ asociată formei explicite la matricea $\bar{\mathscr{C}}$, de asemenea asociată formei explicite, sînt următoarele:

$$\begin{array}{c} \\ x_2 \\ x_4 \\ x_5 \end{array}\left(\begin{array}{cccccc} -2 & 0 & -3 & 0 & 0 & -5 \\ \hline -1 & 1 & 1 & 0 & 0 & 1 \\ 2 & 0 & -1 & 1 & 0 & 2 \\ -1 & 0 & 1 & 0 & 1 & 3 \end{array}\right)$$

$$\Downarrow$$

$$\begin{array}{c} \\ x_2 \\ x_4 \\ x_5 \end{array}\begin{array}{c} \begin{array}{cc} x_1 & x_3 \end{array} \\ \left(\begin{array}{cc|c} -2 & -3 & -5 \\ \hline -1 & 1 & 1 \\ 2 & -1 & 2 \\ -1 & 1 & 3 \end{array}\right) \end{array}$$

$$\Downarrow$$

$$\Downarrow$$

$$\begin{array}{c} \\ x_0 \\ \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{array} \begin{array}{c} \begin{array}{cc} x_1 & x_3 \end{array} \\ \left(\begin{array}{cc|c} -2 & -3 & -5 \\ \hline -1 & 0 & 0 \\ -1 & 1 & 1 \\ 0 & -1 & 0 \\ 2 & -1 & 2 \\ -1 & 1 & 3 \end{array}\right) \end{array} .$$

Observăm că în matricea $\bar{\mathscr{C}}$ putem citi direct pe ultima coloană soluţia de bază corespunzătoare formei explicite, linia zero conţinînd valoarea variabilei x_0, adică valoarea funcţiei obiectiv luată cu semn schimbat, pentru soluţia de bază în chestiune.

Să considerăm problema (P) sub forma explicită (6.6) şi matricea $\bar{\mathscr{C}}$ corespunzătoare formei explicite. Să ne ocupăm acum de modul cum se realizează un pas al algoritmului simplex primal sau algoritmului simplex dual asupra matricei.

Întrucît în trecerea de la matricea $\bar{\mathscr{A}}$ la matricea $\bar{\mathscr{C}}$, de pe linia zero din $\bar{\mathscr{A}}$ s-au suprimat numai elemente nule, iar ultima coloană a suferit doar o rearanjare a elementelor, urmată de introducerea unor elemente nule, condiţiile de admisibilitate primală şi admisibilitate duală şi criteriile de optimalitate ale celor doi algoritmi nu se modifică :

— soluţia de bază de pe ultima coloană a matricei $\bar{\mathscr{C}}$ este primal admisibilă dacă elementele ultimei coloane din $\bar{\mathscr{C}}$ sînt nenegative, posibil cu excepţia celui din linia zero ;

— soluţia de bază de pe ultima coloană a matricei $\bar{\mathscr{C}}$ este dual admisibilă dacă elementele liniei zero din $\bar{\mathscr{C}}$ sînt nenegative, posibil cu excepţia celui din ultima coloană ;

— soluţia de bază de pe ultima coloană a matricei $\bar{\mathscr{C}}$ este o soluţie optimă dacă este în acelaşi timp primal şi dual admisibilă.

Din aceleaşi motive, criteriul de intrare (ieşire) din algoritmul simplex primal (dual) nu suferă modificări :

— criteriul de intrare (ieşire) din algoritmul simplex primal (dual) este : se alege coloana (linia) corespunzătoare elementului minim din linia zero (ultima coloană) din $\bar{\mathscr{C}}$.

Scanned with OKEN Scanner

Observăm că orice coloană din $\bar{\mathscr{A}}$ cu element nenul în linia zero se regăsește în $\bar{\mathscr{C}}$ cu același element nenul în linia zero și elementele pozitive ale coloanei din $\bar{\mathscr{A}}$ se regăsesc în coloana corespunzătoare din $\bar{\mathscr{C}}$, unde nu apar noi elemente pozitive; de asemenea, orice linie din $\bar{\mathscr{A}}$ cu element nenul în ultima coloană se regăsește în $\bar{\mathscr{C}}$ cu același element nenul în ultima coloană și elementele negative ale liniei din $\bar{\mathscr{A}}$ se regăsesc în linia corespunzătoare din $\bar{\mathscr{C}}$, unde nu apar noi elemente negative. Din acestea rezultă că criteriul de existență a unei funcții obiectiv nemărginite inferior în algoritmul simplex primal și criteriul de inexistență a soluțiilor admisibile în algoritmul simplex dual nu suferă modificări:

— dacă în coloana (linia) lui $\bar{\mathscr{C}}$, fixată prin criteriul de intrare (ieșire) al algoritmului simplex primal (dual), toate elementele sînt mai mici (mari) sau egale cu zero, atunci programul liniar nu are soluții optime (admisibile).

Observăm că prin rearanjarea liniilor din $\bar{\mathscr{A}}$ elementele oricărui raport cu numitor pozitiv din coloana fixată se vor regăsi în aceeași linie în $\bar{\mathscr{C}}$ și că nu vor apărea noi astfel de rapoarte; de asemenea, elementele oricărui raport cu numitor negativ din linia fixată se vor regăsi în aceeași coloană în $\bar{\mathscr{C}}$ și că nu vor apărea nici astfel de rapoarte. Din acestea rezultă că criteriul de ieșire (intrare) din algoritmul simplex primal (dual) nu suferă modificări:

— criteriul de ieșire (intrare) din algoritmul simplex primal (dual) este: se alege linia (coloana) din $\bar{\mathscr{C}}$ corespunzătoare raportului minim (maxim) cu numitor pozitiv (negativ) dintre rapoartele formate cu ultima coloană (linia zero) și coloana (linia) fixată.

Pînă acum am arătat că operațiile de aflare a pivotului în algoritmii simpliceali, precum și regulile ce duc la oprirea algoritmilor se realizează asupra matricei $\bar{\mathscr{C}}$ exact în același mod ca asupra matricei $\bar{\mathscr{A}}$. Expresia „nu suferă modificări" trebuie înțeleasă în sensul că folosindu-se o regulă cu același enunț se ajunge prin aplicarea sa asupra matricei $\bar{\mathscr{C}}$ la un rezultat corect.

Să presupunem că s-au aplicat în paralel operațiile unui pas al algoritmului simplex primal sau dual asupra matricelor $\bar{\mathscr{A}}$ și $\bar{\mathscr{C}}$. ajungîndu-se pînă la determinarea pivotului. Am văzut deja că pivoții din cele două matrice sînt identici, aceeași variabilă secundară va deveni principală, aceeași variabilă principală va deveni secun-

Scanned with OKEN Scanner

dară. Este clar că operaţia de pivotaj cu pivotul cunoscut nu poate fi aceeaşi în cele două matrice. Să arătăm modul în care se realizează un pivotaj asupra matricei $\mathscr{C}$, cînd pivotul este dat. Pentru aceasta, vom realiza pivotajul asupra sistemului restricţiilor aflat sub forma (6.6) şi vom arăta că elementele obţinute se regăsesc cînd se aplică matricei $\bar{\mathscr{C}}$ două reguli foarte simple.

Fie α_j^h pivotul şi ecuaţia j din (6.6) scrisă pe larg

$$x_j + \ldots + \alpha_j^h x_h + \ldots + \alpha_j^k x_k + \ldots = \beta_j, \qquad (6.10)$$

unde s-au pus în evidenţă variabila principală, termenul ce conţine pivotul, un termen oarecare şi termenul liber. Împărţind ecuaţia (6.10) cu α_j^h, obţinem ecuaţia h din noul sistem:

$$x_h + \frac{1}{\alpha_j^h} x_j + \ldots + \frac{\alpha_j^k}{\alpha_j^h} x_k + \ldots = \frac{\beta_j}{\alpha_j^h}; \qquad (6.11)$$

deci, coeficienţii ecuaţiei h din noul sistem vor fi

$$\bar{\alpha}_h^j = \frac{1}{\alpha_j^h}, \quad \bar{\alpha}_h^k = \frac{\alpha_j^k}{\alpha_j^h}, \quad \bar{\beta}_h = \frac{\beta_j}{\alpha_j^h}, \quad (k \in K - \{h\}). \qquad (6.12)$$

Considerăm o ecuaţie oarecare i, $(i \neq h)$, din sistemul (6.6) scrisă analog ecuaţiei (6.10):

$$x_i + \ldots + \alpha_i^h x_h + \ldots + \alpha_i^k x_k + \ldots = \beta_i, \qquad (6.13)$$

$$(i \in (\{0, 1, \ldots, n\} - \{h\}).$$

Înmulţim ecuaţia (6.11) cu $-\alpha_i^h$ şi o adunăm membru cu membru la ecuaţia (6.13); obţinem

$$x_i - \frac{\alpha_i^h}{\alpha_j^h} x_j + \ldots + \left(\alpha_i^k - \frac{\alpha_i^h}{\alpha_j^h} \alpha_j^k\right) x_k + \ldots = \beta_i - \frac{\alpha_i^h}{\alpha_j^h} \beta_j, \qquad (6.14)$$

$$(i \in \{0, 1, \ldots, n\} - \{h\}),$$

aşa încît noii coeficienţi ai unei ecuaţii i din sistemul transformat. $(i \neq h)$, vor fi daţi de formulele

$$\bar{\alpha}_i^j = -\frac{\alpha_i^h}{\alpha_j^h}, \quad \bar{\alpha}_i^k = \alpha_i^k - \frac{\alpha_j^k}{\alpha_j^h} \alpha_i^h, \quad (k \in K - \{h\}),$$

$$\bar{\beta}_i = \beta_i - \frac{\beta_j}{\alpha_j^h} \alpha_i^h, \quad (i \in \{0, 1, \ldots, n\} - \{h\}). \qquad (6.15)$$

Scanned with OKEN Scanner

Observăm că pentru $i = j$ se găsesc coeficienţii unei ecuaţii banale, ceea ce era de aşteptat căci variabila x_j a devenit secundară.

Pe baza formulelor (6.12), (6.15) care dau toate elementele matricei obţinute după pivotajul cu pivotul α_j^h vom verifica regulile simple ce realizează pivotajul asupra matricei.

Să considerăm matricea $\bar{\mathscr{C}}$ scrisă sub forma

$$\begin{array}{c|cccc|c} & \dots & x_h & \dots & x_k & \dots & \\ \vdots & & \vdots & & \vdots & & \vdots \\ x_i & \dots & \alpha_i^h & \dots & \alpha_i^k & \dots & \beta_i \\ \vdots & & \vdots & & \vdots & & \vdots \\ x_h & \dots & -1 & \dots & 0 & \dots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \\ x_j & \dots & \alpha_j^h & \dots & \alpha_j^k & \dots & \beta_j \\ \vdots & & \vdots & & \vdots & & \vdots \end{array}$$

unde au fost puse în evidenţă linia şi coloana pivotului, o linie şi o coloană oarecare, linia variabilei ce devine principală şi ultima coloană. Aplicăm acestei matrice operaţiile:

— se împarte coloana pivotului la $-\alpha_j^h$ şi rezultatele se trec în noua matrice într-o coloană cu indicativul x_j în loc de x_h;

— se înmulţeşte noua coloană pe rînd cu toate elementele liniei j din $\bar{\mathscr{C}}$ şi se adună de fiecare dată la coloana de unde au fost luate, trecîndu-se rezultatele în coloanele corespunzătoare din noua matrice. Prin aplicarea acestor operaţii se găseşte matricea

$$\begin{array}{c|cccc|c} & & x_j & & x_k & & \\ \vdots & & \vdots & & \vdots & & \vdots \\ \dots & -\alpha_i^h/\alpha_j^h & \dots & \alpha_i^k - \dfrac{\alpha_i^h}{\alpha_j^h}\,\alpha_j^k & \dots & \beta_i - \dfrac{\beta_j}{\alpha_j^h}\,\alpha_i^h \\ \vdots & & \vdots & & \vdots & & \vdots \\ \dots & 1/\alpha_j^h & \dots & \alpha_j^k/\alpha_j^h & \dots & \beta_j/\alpha_j^h \\ \vdots & & \vdots & & \vdots & & \vdots \\ \dots & -1 & \dots & 0 & \dots & 0 \\ \vdots & & \vdots & & \vdots & & \vdots \end{array}$$

Comparînd elementele matricei cu elementele date de formulele (6.12), (6.15) este evident că operaţiile enunţate realizează tocmai pivotajul sub noua formă.

Exemplul 6.2. Să rezolvăm prin algoritmul simplex primal programul liniar pentru care s-a construit matricea $\overline{\mathscr{C}}$ în exemplul 6.1. Pivotajele succesive sînt următoarele:

x_0	−2	−3	−5	−5	3	−2	5	8	13
x_1	−1	0	0	−1	0	0	1	1	3
x_2	−1	①	1	0	−1	0	0	−1	0
x_3	0	−1	0	−1	1	1	1	2	4
x_4	2	−1	2	①	1	3	−1	0	0
x_5	−1	1	3	0	−1	2	0	−1	2

Soluţia optimală o găsim în ultima coloană din ultima matrice: $x_1 = 3$, $x_2 = 0$, $x_3 = 4$, $x_4 = 0$, $x_5 = 2$ şi valoarea funcţiei obiectiv este $f = -13$.

Exemplul 6.3. Să reconsiderăm exemplul 3.3 şi să rezolvăm programul liniar prin algoritmul simplex dual lucrînd cu noua matrice asociată. Construim în prealabil matricea $\overline{\mathscr{C}}$: în egalitatea (3.26) punem $f = -x_0$ şi o transcriem sub forma

$$x_0 + x_2 + 2x_4 + 3x_5 = 5, \tag{6.16}$$

apoi adăugăm ecuaţiile banale

$$x_2 + (-x_2) = 0, \quad x_4 + (-x_4) = 0, \quad x_5 + (-x_5) = 0 \tag{6.17}$$

şi obţinem sistemul

$$\left.\begin{aligned} x_0 + x_2 + 2x_4 + 3x_5 &= 5 \\ x_1 + x_2 - 2x_4 + x_5 &= -2 \\ x_2 - x_2 &= 0 \\ x_3 - x_2 + x_4 - x_5 &= -3 \\ x_4 - x_1 &= 0 \\ x_5 - x_5 &= 0 \end{aligned}\right\} \tag{6.18}$$

cu matricea asociată

$$\overline{\mathscr{C}} = \left(\begin{array}{cccc} 1 & 2 & 3 & 5 \\ \hline 1 & -2 & 1 & -2 \\ -1 & 0 & 0 & 0 \\ -1 & 1 & -1 & -3 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{array}\right).$$

Pivotajele succesive sînt următoarele:

	x_2	x_4	x_5		x_3	x_4	x_5		x_3	x_1	x_5	
x_0	1	2	3	5	1	3	2	2	4	3	2	−13
x_1	1	−2	1	−2	1	(−1)	0	−5	0	−1	0	0
x_2	−1	0	0	0	−1	−1	1	3	−2	−1	1	8
x_3	(−1)	1	−1	−3	−1	0	0	0	−1	0	0	0
x_4	0	−1	0	0	0	−1	0	0	−1	−1	0	5
x_5	0	0	−1	0	0	0	−1	0	0	0	−1	0

Soluția optimă o găsim în ultima coloană din ultima matrice: $\bar{x}_1 = 0$, $\bar{x}_2 = 8$, $\bar{x}_3 = 0$, $\bar{x}_4 = 5$, $\bar{x}_5 = 0$ și valoarea funcției obiectiv este $= 13$.

6.2. PROBLEMA PROGRAMĂRII LINIARE ÎN NUMERE ÎNTREGI: CAZUL TOTAL ÎNTREG ȘI CAZUL PARȚIAL ÎNTREG

Considerăm problema: să se minimizeze

$$f = c' X, \tag{6.19}$$

în condițiile

$$AX = b, \quad X \geqslant 0, \tag{6.20}$$

și

$$x_h \text{ număr întreg}, \ (h \in \mathscr{R}), \tag{6.21}$$

unde A este o matrice $m \times n$, b și c sînt vectori m-dimensional și n-dimensional, respectiv, iar $\mathscr{R}$ este o mulțime de indici $\mathscr{R} \subset \{1, 2, \ldots, n\}$.

Problema (6.19) — (6.21) nu este o problemă de programare liniară în sensul definiției din § 2.2, datorită prezenței condițiilor (6.21), decît în cazul extrem $\mathscr{R} = \emptyset$. Totuși vom numi această problemă o problemă de programare liniară în numere întregi pentru a păstra denumirea folosită în literatura științifică; desigur, ter-

menul „liniară" este introdus în denumire pentru a releva faptul că funcţia (6.19) şi condiţiile (6.20) sînt liniare. Dacă $\mathcal{R} = \{1, 2, \ldots, n\}$, vom spune că problema (6.19)—(6.21) este un program liniar total întreg; dacă $\mathcal{R} \subset \{1, 2, \ldots, n)\}$, $\mathcal{R} \neq \{1, 2, \ldots, n\}$, vom spune că problema (6.19)—(6.21) este un program liniar parţial întreg sau mixt. Denumirile sînt justificate prin aceea că în primul caz toate variabilele se cer a fi cu valoare întreagă, în timp ce în cazul al doilea această proprietate trebuie să o aibă numai o parte dintre variabile; acestea vor fi denumite constrînse, în timp ce celelalte vor fi denumite neconstrînse. Deoarece condiţiile (6.21) fac parte dintre condiţiile problemei, pentru a le individualiza le vom denumi *condiţiile de integritate*; prin urmare, condiţiile problemei de programare liniară în numere întregi sînt de trei feluri: restricţiile problemei (sistemul liniar), condiţiile de nenegativitate şi condiţiile de integritate. Denumirea de soluţie a problemei va fi întrebuinţată ca şi pînă acum pentru a desemna o soluţie a sistemului liniar, denumirea de soluţie admisibilă pentru o soluţie a sistemului de condiţii (6.20), iar denumirea de soluţie admisibilă întreagă o introducem pentru a desemna o soluţie a sistemului de condiţii (6.20), (6.21). Bineînţeles, o soluţie întreagă este o soluţie a sistemului liniar şi a condiţiilor (6.21), dar nu este numaidecît nenegativă; observăm că definiţia nu exclude posibilitatea ca într-o soluţie întreagă să existe o componentă x_h, $h \in \mathcal{R}$, neîntreagă.

Problemele de programare liniară în numere întregi reprezintă o clasă de probleme foarte importantă pentru practică. De multe ori, deşi sensul economic al variabilelor nu permite întrebuinţarea practică a unei variante în care nu toate valorile variabilelor să fie numere întregi, se consideră eronat că problema este un program liniar obişnuit. În acest caz, a nu adăuga condiţiile de integritate şi a proceda la rotunjirea soluţiei optime a programului liniar obişnuit poate conduce la erori grave. Putem proceda aşa doar dacă ne mulţumim cu o soluţie mai bună decît cea folosită în practica curentă, presupunînd că acesta este cazul. O altă situaţie, mai importantă poate decît cea discutată, care conduce la programe liniare în numere întregi, este aceea a problemelor unde variabilele au valori convenţionale, reprezentînd alternative dintr-o mulţime finită de alternative; astfel, sînt de exemplu cazurile cînd variabilele sau o parte a lor iau numai valorile convenţionale zero sau unu, valori reprezentînd o alternativă şi contrara sa.

Ilustrăm comentariile printr-un model matematic frecvent folosit în practică, anume, modelul problemei de atribuire.

Scanned with OKEN Scanner

Se consideră m produse P_i, $(i = 1, \ldots, m)$, care urmează a fi atribuite la n centre de consum, C_j, $(j = 1, \ldots, n)$ astfel ca fiecare produs să se atribuie cel mult unui centru de consum şi fiecare centru de consum să nu primească mai mult de un produs. Prin atribuirea produsului P_i către centrul de consum C_j se obţine o anumită rentabilitate c_{ij}, $(i = 1, \ldots, m;\ j = 1, \ldots, n)$. Bineînţeles, $c_{ij} = 0$ înseamnă că atribuirea lui P_i către C_j nu are rentabilitate, uneori se pune $c_{ij} = 0$ pentru a evidenţia faptul că această atribuire este imposibilă. Problema de atribuire constă în a se stabili pentru orice produs P_i dacă este atribuit sau nu vreunui centru de consum, iar în caz afirmativ cărui centru C_j, în aşa fel ca rentabilitatea totală să fie maximă. Pentru formularea matematică a problemei se introduc necunoscutele x_{ij}, cu semnificaţia următoare: $x_{ij} = 1$ dacă produsul P_i se atribuie centrului de consum C_j şi $x_{ij} = 0$ în caz contrar. Condiţiile problemei se traduc analitic astfel: să se maximizeze

$$f = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}\, x_{ij}, \tag{6.22}$$

cu restricţiile

$$\sum_{j=1}^{n} x_{ij} \leqslant 1, \ (i = 1, \ldots, m), \tag{6.23}$$

$$\sum_{i=1}^{m} x_{ij} \leqslant 1, \ (j = 1, \ldots, n), \tag{6.24}$$

şi

$$x_{ij} = 0 \text{ sau } 1, \ (i = 1, \ldots, m,\ j = 1, \ldots, n). \tag{6.25}$$

Pentru un i fixat, condiţia (6.23) corespunzătoare exprimă faptul că produsul P_i poate fi atribuit cel mult unui centru C_j, deoarece suma din membrul stîng nu poate avea decît valoarea zero sau unu, în baza condiţiilor (6.25). Pentru un j fixat, condiţia (6.24) corespunzătoare exprimă faptul că i se poate atribui centrului C_j cel mult un produs, deoarece suma din membrul stîng nu poate avea decît valoarea zero sau unu, din acelaşi motiv. Rentabilitatea atribuirii

produsului P_i către centrul C_j este $c_{ij}\,x_{ij}$, căci dacă are loc atribuirea, avem $x_{ij} = 1$ și rentabilitatea este c_{ij}, iar dacă nu are loc, avem $x_{ij} = 0$ și rentabilitatea este nulă; prin urmare, rentabilitatea totală este dată într-adevăr de funcția (6.22). Condițiile (6.25) exprimă definiția variabilelor. Problema (6.22) — (6.25) este un program liniar total întreg, căci sistemul liniar (6.23), (6.24) constituie restricțiile, condițiile (6.25) reprezintă condițiile de integritate, condițiile de nenegativitate sînt o consecință a condițiilor (6.25), deci pot fi omise, iar funcția obiectiv (6.22) este o funcție liniară. Mai mult, acest program liniar total întreg impune condiții de integritate particulare, căci variabilele nu pot lua decît două valori, zero sau unu, și de aceea problema aparține clasei de programe în numere întregi, denumite programe în variabile bivalente sau programe pseudo-booleene.

Problema de atribuire în maniera formulată apare ca o problemă nerealistă dacă dăm un sens prea îngust termenilor de „produs" și „centru de consum" folosiți în enunț, căci un produs se repartizează în general către mai multe centre de consum și fiecare centru primește mai multe produse. Totuși, se pot da exemple de probleme practice pentru care este convenabil modelul problemei de atribuire, dacă înțelegem termenii în sens larg. Astfel, punînd produs = candidat, centru = post și c_{ij} = rentabilitatea angajării candidatului P_i în postul C_j, problema de atribuire reprezintă problema angajării a m candidați în n posturi disponibile, cu rentabilitate totală maximă. Desigur, se presupune că măcar unii dintre candidați au pregătirea necesară pentru a putea ocupa cel puțin două posturi. Pe aceeași cale se poate da o formulare simplificată problemei întocmirii orarului avioanelor și altor probleme. Chiar problema de atribuire ar putea fi formulată și într-o manieră deosebită de cea prezentată mai sus.

Algoritmii pentru rezolvarea programelor liniare în numere întregi sînt de diverse tipuri, după instrumentul matematic folosit în formularea lor. Cel mai mult utilizați par a fi algoritmii de secționare, care dealtfel au fost și primii descoperiți, și algoritmii de ramificare și mărginire. În continuare, vom expune unii algoritmi de aceste două tipuri, atît pentru rezolvarea programelor liniare total întregi, cît și pentru rezolvarea programelor liniare parțial întregi. Cu toate că există și algoritmi specifici pentru rezolvarea programelor liniare în variabile bivalente, nu vom da nici un astfel de algoritm.

Este demn de reținut faptul că în teoria programelor în numere întregi există o remarcabilă contribuție științifică românească.

Scanned with OKEN Scanner

6.3. ALGORITMI DE SECȚIONARE

Un algoritm de secționare rezolvă un program liniar în numere întregi (6.19) — (6.21) în mai multe cicluri de pași. Operațiile mari ale unui ciclu sînt următoarele: (A) Se rezolvă un program liniar obișnuit, obținut din programul în numere întregi disponibil la începutul ciclului, prin omiterea condițiilor de integritate. Aplicarea procedeului va putea continua numai dacă rezolvarea s-a încheiat cu determinarea unei soluții optime, căci în caz contrar rezultă foarte simplu concluzia că programul în numere întregi inițial nu are soluții optime; dacă în soluția optimă determinată toate variabilele x_h, $(h \in \mathscr{R})$, au valori întregi, atunci aceasta este o soluție optimă și pentru programul în numere întregi inițial, procedeul este de asemenea oprit. (B) Dacă în soluția optimă determinată există cel puțin o variabilă x_h, $(h \in \mathscr{R})$, cu o valoare neîntreagă, atunci se adaugă programului considerat la începutul ciclului o restricție suplimentară, astfel construită încît ea nu este satisfăcută de soluția optimă găsită, dar este satisfăcută de orice soluție admisibilă întreagă a programului inițial. Pe această cale programul liniar în numere întregi disponibil la începutul ciclului se înlocuiește cu altul echivalent lui, în sensul că are aceeași mulțime de soluții admisibile întregi, și se trece la aplicarea unui nou ciclu de pași asupra noului program. Se înțelege, toate programele construite succesiv în urma ciclurilor de pași sînt echivalente celui inițial, în sensul explicat.

Deoarece mulțimea soluțiilor admisibile ale programului considerat se împarte mereu în două, prin introducerea unei restricții suplimentare, numai una dintre părți fiind cercetate în continuare, un algoritm cu aceste două operații mari este denumit *algoritm de secționare*. Vom da în cele ce urmează algoritmii de secționare ai lui R. Gomory pentru rezolvarea programelor liniare total întregi și parțial întregi [8, 9]. Operația (A) se realizează cu algoritmul simplex dual sub forma dată în primul paragraf din acest capitol. Avantajul obținut pe această cale va fi explicat mai departe.

Din cele de mai sus rezultă că pentru formularea algoritmilor menționați este necesar să se arate procedeul de generare a restricțiilor suplimentare ce urmează a fi adăugate problemei în cazul cînd după rezolvarea programului liniar obișnuit se obține o soluție optimă neîntreagă. Acest procedeu depinde de faptul dacă programul considerat este total întreg sau parțial întreg, astfel încît

Scanned with OKEN Scanner

va trebui să ne ocupăm pe rînd de cele două cazuri. Oricum, înainte de generarea vreunei restricţii suplimentare, programul în numere întregi considerat, sau un program echivalent cu el, se va afla sub o formă explicită primal şi dual admisibilă ca urmare a încheierii aplicării algoritmului simplex dual. Aşadar, programul liniar în numere întregi obţinut după operaţia (A) este de forma : să se minimizeze

$$f = -x_0, \tag{6.26}$$

în condiţiile

$$x_i + \sum_{k \in K} \alpha_i^k x_k = \beta_i, \ (i = 0, 1, \ldots, n), \tag{6.27}$$

$$x_i \geqslant 0, \ (i = 1, \ldots, n), \tag{6.28}$$

$$x_h \text{ întreg}, \ (h \in \mathscr{R}), \tag{6.29}$$

unde

$$\alpha_0^k \geqslant 0, \ (k \in K), \tag{6.30}$$

$$\beta_i \geqslant 0, \ (i \in N \setminus K), \quad \beta_k = 0, \ (k \in K), \tag{6.31}$$

iar K este mulţimea indicilor variabilelor secundare.

Observăm că s-au omis indicii care ar arăta numărul ciclului tocmai executat, în scopul folosirii notaţiilor celor mai simple posibile.

Soluţia de bază optimă disponibilă este

$$x_i = \beta_i, \ (i \in N \setminus K), \ x_k = 0, \ (k \in K), \tag{6.32}$$

şi ca urmare a ipotezei de lucru există un indice $h \in \mathscr{R} \cap (N \setminus K)$, astfel încît β_h este un număr fracţionar. Acum, trebuie să generăm o restricţie suplimentară care să aibă următoarele două proprietăţi : a) să nu fie satisfăcută de soluţia optimă (6.32), b) să fie satisfăcută de orice soluţie admisibilă întreagă a programului. Desigur, uneori s-ar putea construi mai multe asemenea restricţii, însă algoritmii în chestiune construiesc doar una. O asemenea restricţie

Scanned with OKEN Scanner

este construită plecînd de la o singură ecuaţie din (6.27), anume aceea cu partea fracţionară maximă a membrului drept, în cazul total întreg, sau aceea cu partea fracţionară maximă a membrului drept dintre ecuaţiile conţinînd o variabilă principală constrînsă, în cazul parţial întreg. Existenţa unei asemenea ecuaţii în sistemul (6.27) este asigurată prin ipoteză, dar construirea restricţiei suplimentare folosindu-se numai una dintre ecuaţiile (6.27) şi considerarea drept ecuaţie generatoare a celei cu partea fracţionară a membrului drept maximă sînt particularităţi ale algoritmilor lui R. Gomory, alese numai din necesitatea ca algoritmii să aibă o evoluţie unic determinată. Dealtfel, aşa se explică de ce numărul variantelor algoritmilor de secţionare este foarte mare.

Să considerăm, prin urmare, ecuaţia generatoare determinată, în general, în mod unic de regula de mai sus şi pentru simplitate să o rescriem sub forma

$$y + \sum_{k \in K} \gamma_k x_k = \delta. \tag{6.33}$$

Variabila principală din ecuaţie s-a notat cu y, coeficienţii variabilelor secundare cu γ_k, $(k \in K)$, iar termenul liber cu δ. Reamintim că δ este un număr pozitiv neîntreg; notăm ca de obicei cu $[\delta]$ partea întreagă a lui δ — cel mai mare număr întreg mai mic sau egal cu δ — şi condiţiile satisfăcute de δ sînt $0 < \delta - [\delta] < 1$. Observăm că ecuaţia generatoare ar putea să nu fie unic determinată dacă în sistemul (6.27) există mai multe ecuaţii conţinînd o variabilă principală x_h, $(h \in \mathcal{R})$, cu termenul liber, avînd partea fracţionară maximă. Se obişnuieşte să se ia în acest caz prima ecuaţie întîlnită cu această proprietate.

Să presupunem că programul (6.26) — (6.29) este un program liniar total întreg, $\mathcal{R} = \{1, 2, \ldots, n\}$. Atunci, restricţia suplimentară corespunzătoare se construieşte pe baza teoremei următoare:

Teorema 6.1. *Dacă ecuaţia* (6.33) *este o ecuaţie cu termenul liber neîntreg din sistemul restricţiilor programului* (6.26) — (6.29) *cu* $\mathcal{R} = N$ *şi care satisface condiţiile* (6.30), (6.31) *atunci ecuaţia*

$$\xi - \sum_{k \in K} (\gamma_k - [\gamma_k])\, x_k = -(\delta - [\delta]), \tag{6.34}$$

cu ξ *o variabilă nouă nenegativă şi întreagă, nu este satisfăcută de soluţia optimă* (6.32) *şi este satisfăcută de orice soluţie admisibilă întreagă a programului.*

Scanned with OKEN Scanner

Demonstrație. Dacă înlocuim soluția optimă (6.32) în ecuația (6.34), obținem $\xi = -(\delta - [\delta])$; ultima egalitate împreună cu $\delta > [\delta]$ contrazice $\xi \geqslant 0$, ceea ce înseamnă că ecuația (6.34) nu este satisfăcută de soluția optimă (6.32). Să demonstrăm a doua afirmație a teoremei. Pentru aceasta scriem în prealabil ecuația (6.33) sub forma

$$\sum_{k \in K} (\gamma_k - [\gamma_k])\, x_k - (\delta - [\delta]) = [\delta] - y - \sum_{k \in K} [\gamma_k]\, x_k \qquad (6.35)$$

și notăm

$$\xi = [\delta] - y - \sum_{k \in K} [\gamma_k]\, x_k, \qquad (6.36)$$

apoi facem următorul raționament: oricare ar fi soluția admisibilă întreagă a programului liniar total întreg, substituită în (6.36), va da evident o valoare întreagă pentru ξ; această valoare este și nenegativă, căci dacă $\xi < 0$ și întreg, avem $\xi \leqslant -1$ și aceeași soluție ar trebui să satisfacă inegalitatea

$$\sum_{k \in K} (\gamma_k - [\gamma_k])\, x_k \leqslant -1 + (\delta - [\delta]), \qquad (6.37)$$

dedusă din (6.35), ceea ce este imposibil, membrul stîng fiind nenegativ iar membrul drept negativ. Rezultă că orice soluție admisibilă întreagă a programului liniar total întreg satisface ecuația obținută din (6.35) prin înlocuirea membrului drept cu ξ și dă o valoare întreagă și nenegativă a noii variabile.

Observăm că deși ecuația (6.34) pare a avea un aspect complicat în realitate ea se construiește foarte simplu din ecuația generatoare (6.33), anume: se introduce o nouă variabilă principală întreagă nenegativă ξ, se iau drept coeficienți ai variabilelor secundare părțile fracționare cu semn schimbat ale coeficienților corespunzători din ecuația (6.33) și drept termen liber se ia partea fracționară cu semn schimbat a termenului liber din ecuația (6.33).

Mai observăm că pentru a se adăuga ecuația (6.34) la sistemul restricțiilor programului este foarte comod să folosim matricea asociată descrisă în primul paragraf: din linia ecuației generatoare formăm linia părților fracționare ale elementelor, schimbăm toate

semnele şi linia rezultată o adăugăm în josul matricei. Dacă s-ar folosi matricea asociată obişnuită ar trebui să adăugăm în afară de o nouă linie corespunzătoare restricţiei suplimentare şi o nouă coloană corespunzătoare noii variabile.

Observăm de asemenea că prin adăugarea unei linii la matricea $\overline{\mathscr{C}}$ se va transmite admisibilitatea dual la soluţia de bază din matricea obţinută, dar nu se va transmite admisibilitatea primală, căci termenul liber al restricţiei adăugate este negativ. De aceea, este convenabil să folosim algoritmul simplex dual pentru rezolvarea programelor liniare obţinute succesiv. Prin urmare, vom folosi ca procedeu auxiliar în algoritmul de secţionare, algoritmul simplex dual aplicat sub forma descrisă în primul paragraf din acest capitol.

Exemplul 6.4. Să considerăm programul liniar total întreg dat prin matricea asociată

$$\overline{\mathscr{C}} = \left(\begin{array}{cc|c} 1/5 & 2/5 & 11/5 \\ \hline -1 & 0 & 0 \\ 0 & -1 & 0 \\ 1/5 & -7/5 & 2/5 \\ 2/5 & 1/5 & 1/5 \\ -3/5 & 1/5 & 7/5 \end{array}\right).$$

Soluţia de bază corespunzătoare este $x_1 = 0$, $x_2 = 0$, $x_3 = 2/5$, $x_4 = 1/5$, $x_5 = 7/5$ şi este o soluţie optimă pentru programul liniar obişnuit, obţinut din cel dat omiţînd condiţiile de integritate. Cum această soluţie nu este întreagă, vom adăuga programului o restricţie suplimentară. Părţile fracţionare ale termenilor liberi sînt 0, 0, 2/5, 1/5, 2/5; Valoarea maximă fiind 2/5, vom lua drept ecuaţie generatoare ecuaţia corespunzătoare liniei a treia, adică ecuaţia

$$x_3 + \frac{1}{5}x_1 - \frac{7}{5}x_2 = \frac{2}{5}, \tag{6.38}$$

deoarece aceasta este prima întîlnită dintre cele cu partea fracţionară a termenului liber egală cu valoarea maximă 2/5. Cum

$$\begin{gathered} [1/5] = 0, \quad [-7/5] = -2, \quad [2/5] = 0 \\ 1/5 - [1/5] = 1/5, \quad -7/5 - [-7/5] = 3/5, \quad 2/5 - [2/5] = 2/5, \end{gathered} \tag{6.39}$$

rezultă din (6.34) că restricţia suplimentară este

$$\xi - \frac{1}{5} x_1 - \frac{3}{5} x_2 = -\frac{2}{5}; \tag{6.40}$$

linia corespunzătoare acestei ecuaţii trebuie adăugată matricei $\overline{\mathscr{C}}$. Dealtfel, pentru a scrie noua matrice asociată unui program liniar total întreg echivalent cu cel dat, nu era necesar să scriem separat ecuaţiile (6.38) sau (6.40), căci completarea matricei cu o nouă linie se putea face direct; oricum, s-a obţinut matricea

$$\overline{\mathscr{C}} = \left(\begin{array}{cc|c} 1/5 & 2/5 & 11/5 \\ \hline -1 & 0 & 0 \\ 0 & -1 & 0 \\ 1/5 & -7/5 & 2/5 \\ 2/5 & 1/5 & 1/5 \\ -3/5 & 1/5 & 7/5 \\ -1/5 & -3/5 & -2/5 \end{array}\right).$$

Acum, pentru a se continua rezolvarea se aplică algoritmul smiplex dual, căci aşa cum observasem mai sus soluţia de bază disponibilă $x_1 = 0$, $x_2 = 0$, $x_3 = 2/5$, $x_4 = 1/5$, $x_5 = 7/5$, $\xi = -2/5$ este dual admisibilă, fără a fi primal admisibilă. Prin urmare, va trebui să tratăm noul program liniar total întreg ca pe cel iniţial, cu deosebirea că acum nu vom găsi soluţia optimă chiar în tabel, cum s-a întîmplat mai înainte, ci va trebui s-o calculăm prin algoritmul simplex dual. Dacă se va obţine o soluţie optimă şi ea nu va fi întreagă, rezultat la care va ajunge cititorul care va continua calculele, atunci se va adăuga o nouă restricţie suplimentară şi se va relua procedeul pentru un nou program liniar total întreg.

Să presupunem că programul (6.26) — (6.29) este un program liniar parţial întreg, $\mathscr{R} \subset N$, $\mathscr{R} \neq N$. Atunci, restricţia suplimentară corespunzătoare se construieşte pe baza teoremei următoare:

Teorema 6.2. *Dacă ecuaţia* (6.33) *este o ecuaţie cu o variabilă principală constrînsă şi cu termen liber neîntreg, din sistemul restricţiilor programului* (6.26) — (6.29) *cu* $\mathscr{R} \subset N$, $\mathscr{R} \neq N$ *şi care satisface condiţiile* (6.30), (6.31), *atunci ecuaţia*

$$\xi - \sum_{k \in K_+} \gamma_k x_k + \frac{\delta - [\delta]}{1 - (\delta - [\delta])} \cdot \sum_{k \in K_-} \gamma_k x_k = -(\delta - [\delta]) \tag{6.41}$$

cu ξ *o variabilă nouă nenegativă şi*

$$K_+ = \{k \mid k \in K,\ \gamma_k > 0\},\ K_- = \{k \mid k \in K,\ \gamma_k < 0\}, \tag{6.42}$$

nu este satisfăcută de soluția optimă (6.32) *și este satisfăcută de orice soluție admisibilă întreagă a programului.*

Demonstrație. Prima afirmație a teoremei se justifică exact ca în teorema precedentă, așa încît rămîne să demonstrăm a doua afirmație. Pentru aceasta scriem ecuația (6.33) sub forma

$$\sum_{k\in K_+} \gamma_k x_k + \sum_{k\in K_-} \gamma_k x_k - (\delta - [\delta]) = [\delta] - y, \tag{6.43}$$

apoi facem următorul raționament: cum y este o variabilă constrînsă, pentru orice soluție admisibilă întreagă a programului membrul drept este un număr întreg, deci are loc sigur una dintre inegalitățile:

$$\sum_{k\in K_+} \gamma_k x_k + \sum_{k\in K_-} \gamma_k x_k - (\delta - [\delta]) \geqslant 0 \tag{6.44}$$

sau

$$\sum_{k\in K_+} \gamma_k x_k + \sum_{k\in K_-} \gamma_k x_k - (\delta - [\delta]) \leqslant -1; \tag{6.45}$$

dacă ținem seamă de admisibilitatea soluției și de semnele coeficienților, rezultă că are loc cu atît mai mult una dintre inegalitățile:

$$\sum_{k\in K_+} \gamma_k x_k \geqslant \delta - [\delta] \tag{6.46}$$

sau

$$\sum_{k\in K_-} \gamma_k x_k \leqslant -1 + (\delta - [\delta]), \tag{6.47}$$

dintre care a doua se poate transcrie

$$-\frac{\delta - [\delta]}{1 - (\delta - [\delta])} \sum_{k\in K_-} \gamma_k x_k \geqslant \delta - [\delta]; \tag{6.48}$$

Scanned with OKEN Scanner

deoarece primii membri din inegalităţile (6.46), (6.48) sînt nenegativi, rezultă că pentru orice soluţie admisibilă întreagă a programului avem

$$\sum_{k \in K_+} \gamma_k x_k - \frac{\delta - [\delta]}{1 - (\delta - [\delta])} \cdot \sum_{k \in K_-} \gamma_k x_k \geqslant \delta - [\delta], \qquad (6.49)$$

iar această inecuaţie se poate scrie sub forma (6.41), introducînd o variabilă nouă nenegativă ξ.

Observăm că ecuaţia (6.41) este ceva mai complicată decît ecuaţia (6.34); ea se construieşte din ecuaţia generatoare (6.33) astfel: se introduce o variabilă nouă nenegativă ξ, termenilor cu coeficienţi pozitivi li se schimbă semnul, termenii cu coeficienţi negativi se înmulţesc toţi cu numărul

$$\frac{\delta - [\delta]}{1 - (\delta - [\delta])}, \qquad (6.50)$$

iar drept termen liber se ia partea fracţionară cu semn schimbat a termenului liber din ecuaţia (6.33).

Mai observăm că şi de această dată este comod să folosim pentru adăugarea ecuaţiei (6.41) la sistemul restricţiilor programului, tot matricea asociată descrisă în primul paragraf, unde din linia ecuaţiei generatoare formăm linia adăugată.

Observăm de asemenea că şi în acest caz după adăugarea liniei la matricea asociată formei explicite a programului liniar se va menţine admisibilitatea dual, dar nu se va menţine admisibilitatea primală a soluţiei de bază din matrice, aşa încît va fi convenabil şi acum să folosim algoritmul simplex dual pentru rezolvarea programelor liniare obişnuite succesive.

Exemplul 6.5. Să considerăm programul liniar parţial întreg dat prin matricea asociată $\overline{\mathscr{C}}$ din exemplul precedent şi precizarea că $\mathscr{R} = \{1, 3, 4\}$. Soluţia de bază disponibilă în matrice, soluţie optimă pentru programul liniar obişnuit obţinut din cel dat prin omiterea condiţiilor de integritate, nu are valori întregi pentru variabilele constrînse x_3 şi x_4. De aceea, va trebui să adăugăm o restricţie suplimentară generată dintr-una din ecuaţiile ce conţin pe x_3 sau x_4 drept variabilă principală. Dintre acestea cea cu partea fracţionară maximă este ecuaţia (6.38), deci ea va fi folosită ca ecuaţie generatoare. Rezultatele din (6.39) permit determinarea numărului (6.50) egal cu 2/3, astfel că putem scrie restricţia suplimentară

$$\xi - \frac{1}{5} x_1 - \frac{14}{15} x_2 = -\frac{2}{5}. \qquad (6.51)$$

Scanned with OKEN Scanner

Completînd matricea $\overline{\mathscr{C}}$ cu linia corespunzătoare acestei ecuaţii, se obţine matricea

$$\overline{\mathscr{C}}'' = \left(\begin{array}{cc|c} 1/5 & 2/5 & 11/5 \\ \hline -1 & 0 & 0 \\ 0 & -1 & 0 \\ 1/5 & -7/5 & 2/5 \\ 2/5 & 1/5 & 1/5 \\ -3/5 & 1/5 & 7/5 \\ -1/5 & -14/15 & -2/5 \end{array}\right).$$

Pentru a se continua rezolvarea se aplică algoritmul simplex dual.

Să trecem la formularea algoritmilor de secţionare ai lui R. Gomory, pentru rezolvarea unui program liniar în numere întregi; vom formula împreună cei doi algoritmi, indicînd numai operaţiile mari ale algoritmilor.

Să presupunem că s-au executat deja r cicluri de paşi ai algoritmului pentru rezolvarea unui program liniar total (parţial) întreg, ($r \geqslant 0$), şi că în cazul $r > 0$ dispunem de un program liniar total (parţial) întreg echivalent celui iniţial. Operaţiile mari ale unui nou ciclu de paşi sînt următoarele.

1. Se rezolvă prin algoritmul simplex dual programul liniar obişnuit obţinut din programul liniar în numere întregi disponibil prin omiterea condiţiilor de integritate: dacă nu se obţine o soluţie optimă, programul liniar în numere întregi nu are soluţii optime; dacă se obţine o soluţie optimă întreagă, aceasta este o soluţie optimă a programului liniar în numere întregi; dacă se obţine o soluţie optimă neîntreagă, trecem la operaţia următoare.

2. Se determină în forma explicită finală a programului, dintre toate ecuaţiile ce conţin o variabilă principală constrînsă, aceea care are partea fracţionară maximă a termenului liber; dacă există mai multe ecuaţii cu aceeaşi parte fracţionară maximă a termenului liber, se reţine prima întîlnită.

3. Se lărgeşte matricea asociată formei explicite finale a programului, introducîndu-se şi o restricţie suplimentară, construită din ecuaţia determinată la punctul precedent pe baza formulei (6.34),

Scanned with OKEN Scanner

[(6.41)], dacă programul liniar este total (parţial) întreg; apoi se trece la un nou ciclu de paşi.

Observăm că în primul ciclu de paşi operaţia 1 se poate realiza asupra matricei asociate $\bar{\mathscr{C}}$, chiar dacă programul liniar nu este iniţial sub o formă explicită dual admisibilă, dar necesită ca de obicei două faze; rezolvarea cerută în operaţia 1 s-ar putea dealtfel realiza în primul ciclu de paşi şi cu ajutorul algoritmului simplex primal, dacă dispunem iniţial de o formă explicită primal admisibilă a programului. În orice caz începînd cu al doilea ciclu de paşi, se foloseşte algoritmul simplex dual, aplicat matricei asociate a noului program liniar.

Algoritmii sînt justificaţi prin teoremele 6.1 şi 6.2, care afirmă că toate programele liniare în numere întregi obţinute succesiv au aceeaşi mulţime a soluţiilor admisibile întregi ca programul iniţial. Se demonstrează (v. [2], § 8.10, § 9.3) că algoritmii sînt convergenţi, dacă se iau unele precauţii suplimentare ce modifică destul de puţin operaţiile unui ciclu de paşi expuse mai înainte. În practică modificările respective nu sînt de obicei introduse, deoarece probabilitatea considerării unui program pentru care convergenţa nu-i asigurată fără regulile suplimentare pomenite este destul de mică.

Un fapt ce ar putea provoca îndoieli cu privire la convergenţa celor doi algoritmi este acela că, aparent, prin adăugarea de restricţii suplimentare, matricea s-ar lărgi necontenit. Or, în realitate, nu se pot adăuga mai mult decît $n - m$ restricţii suplimentare independente în variabilele problemei. În practică, ori de cîte ori o variabilă introdusă printr-o restricţie suplimentară redevine principală, linia corespunzătoare este chiar omisă, căci reprezintă o combinaţie a acelei restricţii cu celelalte restricţii suplimentare şi cu ecuaţiile sistemului, iar restricţia respectivă nu mai intră în nici o altă combinaţie. Astfel, pot rămîne în problemă numai atîtea variabile noi, cît este numărul variabilelor secundare, adică $n - m$. Matricea problemei nu va putea avea, prin urmare, mai mult de $2n - m + 1$ linii, atunci cînd se suprimă sistematic liniile corespunzătoare noilor variabile ce au redevenit principale.

Întrucît algoritmii prezentaţi conţin ca procedeu auxiliar algoritmul simplex dual, în care se fac împărţiri de numere, şi chiar la construirea restricţiilor suplimentare se fac împărţiri, calculul poate produce erori mari, fapt foarte important, căci este necesar la sfîrşitul operaţiei întîi din fiecare ciclu de paşi să ne convingem dacă soluţia optimă găsită este întreagă sau nu. Or, datorită erorilor,

Scanned with OKEN Scanner

s-ar putea ca o soluţie optimă întreagă să nu apară întreagă. De aceea, R. Gomory a dat şi un alt algoritm de secţionare în care se lucrează mereu cu o matrice întreagă, nefăcîndu-se împărţiri. Dealtfel, una dintre superiorităţile algoritmilor de ramificare şi mărginire faţă de algoritmii de secţionare este aceea că primii nu conţin împărţiri.

6.4. ALGORITMI DE RAMIFICARE ŞI MĂRGINIRE

Algoritmii de ramificare şi mărginire presupun cunoaşterea cîtorva noţiuni simple din teoria grafurilor; de aceea, înainte de a începe prezentarea algoritmilor este necesar să introducem conceptele acestei teorii, ce vor fi folosite în continuare.

Se numeşte *graf finit orientat* cuplul format dintr-o mulţime finită V şi o mulţime U de perechi ordonate de elemente din V. În continuare vom folosi denumirea de graf numai în acest sens şi vom nota un graf prin $G = (V, U)$. Orice element din V se numeşte *vîrf*, iar orice element din U se numeşte *arc* al grafului. Denumirile provin din posibilitatea de a reprezenta geometric un graf, figurînd elementele din V ca puncte în plan, iar perechile ordonate din U ca segmente orientate ce leagă punctele respective. Într-un graf, două vîrfuri oarecare pot fi legate sau nu printr-un arc, iar dacă sînt legate, legătura este realizată fie de un arc, fie de două arce cu sensuri opuse; nu vom admite mai multe arce de acelaşi sens între două vîrfuri. Se numeşte *drum* în graf de la vîrful a la vîrful b al grafului o succesiune alternată de vîrfuri şi arce, în care orice arc are drept extremitate iniţială vîrful ce-l precede şi drept extremitate finală vîrful ce-l succede, iar primul vîrf este a şi ultimul este b. Se numeşte *arborescenţă* un graf G în care există un vîrf v numit rădăcină, astfel că de la v există cîte un drum şi numai unul la oricare alt vîrf din graf. Am introdus conceptul de arborescenţă, ilustrat prin reprezentarea geometrică din figura 1, deoarece un algoritm de ramificare şi mărginire construieşte totdeauna o arborescenţă asociată programului liniar în numere întregi. Mersul algoritmului este reprezentat intuitiv prin parcurgerea tuturor nodurilor arborescenţei, sau numai a unei părţi a mulţimii nodurilor, în cazul în care este justificat de o anumită operaţie a algoritmului,

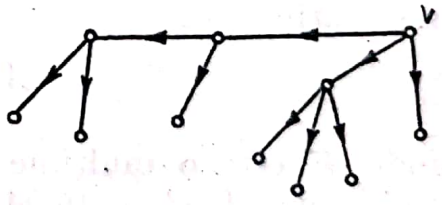


Fig. 1

ca un nod împreună cu cele care-l succed, ori mai multe asemenea subarborescențe, să nu fie parcurse.

Există mulți algoritmi de ramificare și mărginire, atît pentru rezolvarea problemelor de programare liniară în numere întregi, cît și pentru rezolvarea unor alte clase de probleme. În cele ce urmează vom da numai un singur algoritm, dintre cei considerați mai eficienți, anume algoritmul lui B. Korte [10], arătînd mai întîi cum se formulează acesta pentru a servi la rezolvarea programelor liniare total întregi cu variabile mărginite și apoi cum poate fi modificat pentru a fi folosit la rezolvarea programelor liniare parțial întregi.

Considerăm programul liniar total întreg : să se minimizeze

$$f = c'X, \tag{6.52}$$

în condițiile

$$AX \leqslant b,\ X \geqslant 0,\ X \text{ fiind un vector întreg,} \tag{6.53}$$

și

$$X \leqslant d, \tag{6.54}$$

unde A este o matrice de tip $m \times n$ și b, c, d sînt vectori de dimensiuni m, n, n respectiv. Observăm că în afară de condițiile obișnuite, pentru vectorul X se mai impun condițiile (6.54), care reprezintă condiții de mărginire superioară pentru variabilele problemei; de aceea, problema considerată se va numi program liniar total întreg cu variabile mărginite. Dealtfel, algoritmul ce va fi formulat poate fi folosit exact sub aceeași formă pentru rezolvarea unei probleme mai generale, în care condițiile (6.53), (6.54) sînt înlocuite prin

$$AX \leqslant b, \quad X \in \mathscr{F}, \tag{6.55}$$

unde $\mathscr{F}$ este o mulțime finită. Remarcăm ca un caz particular al problemei (6.52) — (6.54) cazul programelor liniare cu variabile bivalente, ce se obține cînd vectorul d este un vector cu toate componentele egale cu unu.

Pentru cele ce urmează vom nota

$$f_i = A_iX - b_i, \quad (i = 1, \ldots, m), \tag{6.56}$$

Scanned with OKEN Scanner

unde A_i, $(i = 1, \ldots, m)$, sînt liniile matricei A, iar b_i, $(i = 1, \ldots, m)$, sînt componentele vectorului b. Cu aceste notaţii putem transcrie programul liniar total întreg: să se minimizeze

$$f(x_1, \ldots, x_n) \tag{6.57}$$

în condiţiile

$$f_i(x_1, \ldots, x_n) \leqslant 0, \quad (i = 1, \ldots, m), \tag{6.58}$$

$$0 \leqslant x_k \leqslant d_k, \qquad (k = 1, \ldots, n), \tag{6.59}$$

$$x_k \text{ întreg} \qquad (k = 1, \ldots, n); \tag{6.60}$$

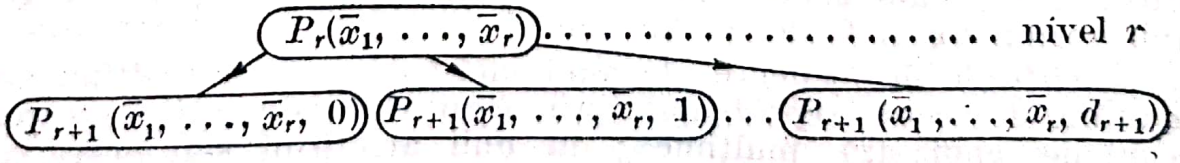
vom folosi aceste notaţii pentru a arăta că algoritmul funcţionează chiar în cazuri mai generale decît cel liniar. În prealabil vom defini arborescenţa asociată problemei (6.57) — (6.60).

Arborescenţa asociată programului liniar total întreg este o arborescenţă în care vîrfurile sînt aşezate pe $n + 1$ nivele, numerotate începînd cu nivelul rădăcinii, căruia îi vom spune nivelul zero; rădăcina va reprezenta mulţimea vectorilor n-dimensionali care satisfac condiţiile (6.59), (6.60). Un vîrf de pe nivelul r, $(0 < r < n)$, reprezintă o mulţime de vectori n-dimensionali, ale căror componente $x_1, \ldots, x_r$ au valori întregi fixate $\bar{x}_1, \ldots, \bar{x}_r$, satisfăcînd (6.59), (6.60), iar celelalte componente nefixate satisfac şi ele condiţiile (6.59), (6.60); notăm

$$P_r(\bar{x}_1, \ldots, \bar{x}_r) = \{X \mid X = \text{vector întreg}; \; x_k = \bar{x}_k, \; (k = 1, \ldots, r); \tag{6.61}$$

$$0 \leqslant x_k \leqslant d_k, \; (k = r + 1, \ldots, n)\}.$$

Din acest vîrf pleacă $d_{r+1} + 1$ arce către tot atîtea vîrfuri ale nivelului $r + 1$, reprezentînd mulţimi de vectori n-dimensionali întregi, în care pe lîngă cele r componente cu valori deja fixate vom avea şi componenta x_{r+1} fixată la una dintre valorile $0, 1, \ldots, d_{r+1}$; anume, valorile sale sînt atribuite pe vîrfuri de la stînga la dreapta astfel:

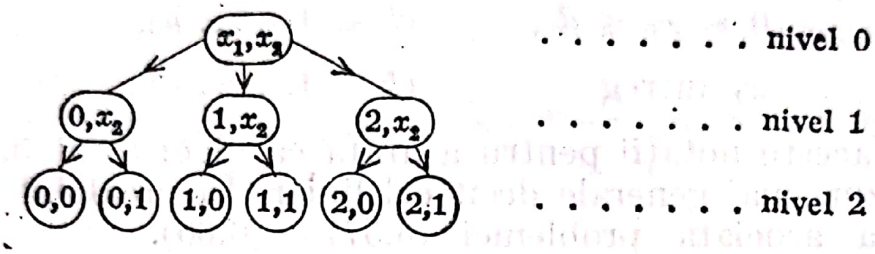


Se înţelege, vîrfurile de pe nivelul n reprezintă mulţimi cu cîte un singur element, anume un vector dintre cei ce satisfac condiţiile (6.59), (6.60); aceste vîrfuri nu au vîrfuri descendente.

Exemplul 6.6. Pentru un program liniar total întreg cu două variabile x_1, x_2, satisfăcînd între altele şi condiţiile

$$0 \leqslant x_1 \leqslant 2,\ 0 \leqslant x_2 \leqslant 1, \tag{6.62}$$

arborescenţa asociată este următoarea:



În interiorul fiecărui vîrf este scrisă forma vectorilor, a căror mulţime este reprezentată prin acel vîrf.

Observăm că a trece de la un vîrf de pe nivelul r la un vîrf de pe nivelul $r+1$ înseamnă a fixa o valoare anumită pentru variabila x_{r+1}, iar a se întoarce de la un vîrf de pe nivelul $r+1$ la un vîrf de pe nivelul r înseamnă a renunţa la valoarea fixată a variabilei x_{r+1}.

Mai observăm că, datorită definiţiei arborescenţei asociate programului liniar total întreg, vectorii reprezentaţi prin vîrfurile nivelului n, considerate de la stînga spre dreapta, sînt în ordine lexicografică. Deoarece algoritmul de care ne ocupăm explorează vîrfurile arborescenţei de sus în jos şi de la stînga spre dreapta, algoritmului i se spune algoritm lexicografic.

Arborescenţa definită pentru a se da un suport intuitiv algoritmului lui B. Korte are multe caracteristici comune cu arborescenţele definite în oricare algoritm de ramificare şi mărginire pentru programe total întregi:

— orice vîrf reprezintă o mulţime de vectori cu anumite componente fixate iar celelalte libere; aceşti vectori sînt uneori soluţii, alteori soluţii admisibile; în unele cazuri, cum este şi al nostru, unii vectori nu sînt nici măcar soluţii;

— vîrfurile descendente ale unui vîrf figurează submulţimi ale mulţimii de vectori reprezentate prin acest vîrf, submulţimi a căror reuniune epuizează mulţimea; în unii algoritmi s-ar putea ca

unele dintre mulţimile corespunzătoare vîrfurilor descendente să fie vide;

— vîrfurile fără descendenţi reprezintă mulţimi cu un singur element; în unii algoritmi acest element este o soluţie admisibilă, întreagă.

Trecerea de la o mulţime corespunzătoare unui vîrf la una dintre mulţimile corespunzătoare unui vîrf descendent este denumită de obicei ramificare, avîndu-se în vedere tocmai suportul intuitiv al algoritmului. În unii algoritmi variabilele nu sînt asociate nivelelor ca în cazul nostru, de aceea apare chestiunea suplimentară de a se alege variabila după care se face ramificarea.

Să trecem acum la alt element al algoritmului: marginile asociate vîrfurilor arborescenţei. Pentru aceasta ne sînt necesare două rezultate simple.

Teorema 6.3. *Fie date numerele* $\bar{x}_1, \ldots, \bar{x}_r$ *satisfăcînd condiţiile* (6.59), (6.60); *dacă există un indice* i *astfel ca*

$$\min f_i(\bar{x}_1, \ldots, \bar{x}_r, x_{r+1}, \ldots, x_n) > 0, \tag{6.63}$$

unde minimul se ia pe mulţimea de vectori $P_r(\bar{x}_1, \ldots, \bar{x}_r)$ *definită prin* (6.61), *atunci în această mulţime nu există nici o soluţie admisibilă întreagă a programului liniar total întreg.*

Demonstraţie. Într-adevăr, dacă ar exista $\bar{X} \in P_r(\bar{x}_1, \ldots, \bar{x}_r)$ care să fie soluţie admisibilă, ar trebui să avem $f_i(\bar{X}) \leqslant 0$, ceea ce ar contrazice ipoteza (6.63).

Teorema 6.4. *Fie date numerele* $\bar{x}_1, \ldots, \bar{x}_r$ *satisfăcînd condiţiile* (6.59), (6.60) *şi numărul* S, *o valoare disponibilă a funcţiei obiectiv* (6.57); *dacă*

$$\min f(\bar{x}_1, \ldots, \bar{x}_r, x_{r+1}, \ldots, x_n) \geqslant S, \tag{6.64}$$

unde minimul se ia pe mulţimea de vectori $P_r(\bar{x}_1, \ldots, \bar{x}_r)$ *definită prin* (6.61), *atunci în această mulţime nu există nici o soluţie admisibilă întreagă „mai bună" decît cea disponibilă.*

Demonstraţie. Într-adevăr, dacă ar exista $\bar{X} \in P_r(\bar{x}_1, \ldots, \bar{x}_r)$ care să fie o soluţie admisibilă întreagă cu $f(\bar{X}) < S$, ar fi contrazisă ipoteza (6.64).

Scanned with OKEN Scanner

Cele două rezultate definesc pentru orice vîrf al arborescenței asociate programului un vector al marginilor, anume, dacă notăm

$$f^r(\bar{x}_1, \ldots, \bar{x}_r) = \min f(\bar{x}_1, \ldots, \bar{x}_r, x_{r+1}, \ldots, x_n),$$
$$f_i^r(\bar{x}_1, \ldots, \bar{x}_r) = \min f_i(\bar{x}_1, \ldots, \bar{x}_r, x_{r+1}, \ldots, x_n), \quad (6.65)$$
$$(i = 1, \ldots, m)$$

valorile minime definite în enunțurile teoremelor 6.3, 6.4 și considerăm vîrful de pe nivelul r căruia îi corespunde mulțimea de vectori n-dimensionali $P_r(\bar{x}_1, \ldots, \bar{x}_r)$, atunci asociem acestui vîrf vectorul $m+1$-dimensional al marginilor

$$(f^r(\bar{x}_1, \ldots, \bar{x}_r), f_1^r(\bar{x}_1, \ldots, \bar{x}_r), \ldots, f_m^r(\bar{x}_1, \ldots, \bar{x}_r)). \quad (6.66)$$

În orice algoritm de ramificare și mărginire se asociază fiecărui vîrf al arborescenței fie o margine, de obicei pentru valorile funcției obiectiv pe mulțimea corespunzătoare acelui vîrf, fie un vector al marginilor, așa cum s-a procedat mai sus, care dă pe lîngă o margine pentru valorile funcției obiectiv și evaluări ale satisfacerii restricțiilor problemei pe mulțimea corespunzătoare vîrfului. De aici provine dealtfel a doua parte a denumirii algoritmilor de acest tip.

Ideea centrală a algoritmilor de ramificare și mărginire este de a se „parcurge", începînd cu rădăcina, pînă la epuizare, mulțimea vîrfurilor arborescenței asociată programului liniar total întreg, în sensul cercetării succesive a tuturor vîrfurilor, unele subarborescențe fiind înlăturate totuși în cursul cercetării pe baza unor rezultate cu privire la marginile asociate vîrfurilor.

În algoritmul lui B. Korte se procedează astfel: se presupune cunoscută o soluție admisibilă întreagă curentă și valoarea corespunzătoare S a funcției obiectiv (inițial soluția curentă este arbitrară iar lui S i se atribuie o valoare suficient de mare); se presupune atins un anumit vîrf al arborescenței, situat pe nivelul r, căruia îi corespunde mulțimea de vectori $P_r(\bar{x}_1, \ldots, \bar{x}_r)$; acum, un pas constă din calculul vectorului marginilor asociat vîrfului și stabilirea uneia dintre următoarele concluzii:

a) dacă prima componentă a vectorului marginilor este cel puțin egală cu S sau una dintre celelalte componente este pozitivă, atunci:

a_1) algoritmul este oprit cînd arborescența este epuizată;

a_2) în caz contrar, ne întoarcem la ultimul vîrf deja întîlnit de pe un nivel superior, de unde pleacă arce pe care încă nu le-am parcurs, şi deplasîndu-ne înainte pe primul arc neparcurs din stînga atingem un nou vîrf, apoi trecem la pasul următor;

b) dacă prima componentă a vectorului marginilor este mai mică decît S şi toate celelalte sînt nepozitive, atunci:

b_1) dacă vîrful cercetat este pe ultimul nivel, schimbăm soluţia curentă şi numărul S prin vectorul corespunzător vîrfului şi, respectiv, prima componentă a vectorului marginilor, iar apoi procedăm ca la a_1) sau a_2) după caz;

b_2) dacă vîrful cercetat nu este pe ultimul nivel, vom avansa la un vîrf descendent, deplasîndu-ne pe primul arc din stînga, şi trecem la pasul următor.

După epuizarea vîrfurilor arborescenţei, dacă S, are valoarea iniţială, programul liniar total întreg nu are soluţii admisibile întregi, iar dacă S are o altă valoare, soluţia curentă este o soluţie optimă a programului şi S valoarea optimă a funcţiei obiectiv.

Algoritmul este justificat prin teoremele 6.3, 6.4 care arată că este inutilă, în cazul a), avansarea la un vîrf descendent, căci fie nu vom găsi o soluţie admisibilă întreagă „mai bună" decît soluţia curentă, fie nu vom găsi o soluţie admisibilă întreagă a programului. Deoarece arborescenţa are un număr finit de vîrfuri, convergenţa algoritmului este justificată. Singura problemă ce mai trebuie clarificată este cum calculăm vectorul marginilor. În cazul unui program liniar total întreg calculul minimului unei funcţii liniare

$$g(x_1, \ldots, x_n) = \sum_{k=1} g_k x_k, \tag{6.67}$$

pe mulţimea $P_r(\bar{x}_1, \ldots, \bar{x}_r)$, se face pe baza formulei evidente

$$\min g(\bar{x}_1, \ldots, \bar{x}_r, x_{r+1}, \ldots, x_n) = \sum_{k=1}^{r} g_k \bar{x}_k + g_0, \tag{6.68}$$

unde g_0 este suma produselor $g_k d_k$ pentru acei indici k dintre $r+1, \ldots, n$, pentru care coeficienţii g_k sînt negativi. Astfel, componentele vectorului marginilor se calculează folosindu-se succesiv formula (6.68) pentru funcţiile $f, f_1, \ldots, f_m$. Observăm că algoritmul permite şi rezolvarea unor programe neliniare total întregi, cu condiţia de a se putea calcula vectorul marginilor.

Scanned with OKEN Scanner

În orice algoritm de ramificare şi mărginire parcurgerea arborescenţei se face pe baza acelorași elemente ca mai sus : o regulă de ramificare, criterii de respingere şi condiţii de oprire. Regula de ramificare arată ce variabilă se va fixa pentru a se realiza trecerea la un vîrf descendent, criteriile de respingere arată că este necesară revenirea la un vîrf anterior, iar condiţiile de oprire semnalează epuizarea arborelui. Se înţelege, eficienţa algoritmului este cu atît mai mare, cu cît numărul vîrfurilor eliminate prin criteriile de respingere este mai mare, pentru anumite clase de probleme.

Exemplul 6.7. Să considerăm următorul program liniar total întreg : să se minimizeze

$$f = x_1 + x_2 - 2x_3, \tag{6.69}$$

în condiţiile

$$f_1 = 2x_1 - x_2 + x_3 \leqslant 0, \quad x_1, x_2, x_3 \in \{0, 1\}. \tag{6.70}$$

Se ia iniţial $S = \max f = 2$ şi un vector arbitrar ca soluţie curentă. Paşii algoritmului ne conduc la considerarea părţii arborescenţei asociate programului, date în fig. 2.

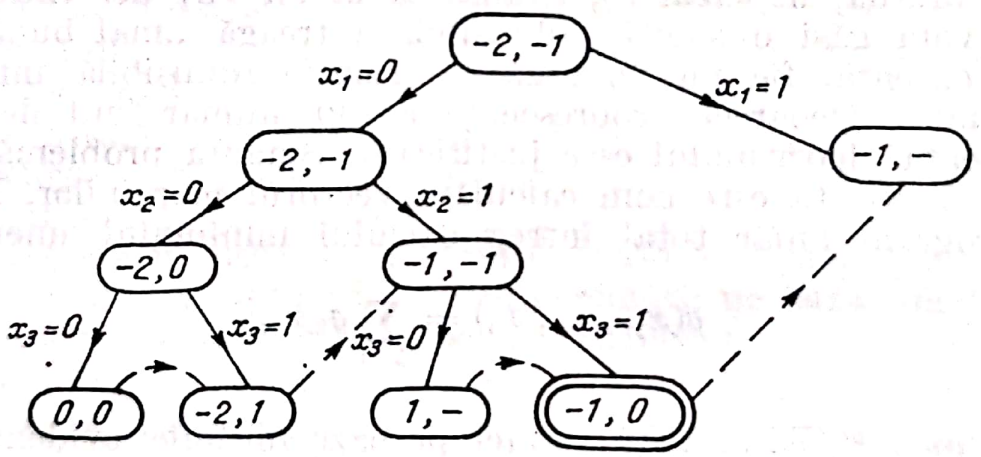


Fig. 2

Perechile de numere scrise în interiorul vîrfurilor reprezintă valorile componentelor vectorului marginilor; acolo unde lipsesc unele componente calculul lor n-a mai fost necesar. Arcele duse cu linie punctată indică mersul înapoi. S-au obţinut : după patru paşi, soluţia admisibilă întreagă $x_1 = 0$, $x_2 = 0$, $x_3 = 0$ cu $f = 0$; după alţi patru paşi, soluţia admisibilă întreagă $x_1 = 0$, $x_2 = 1$, $x_3 = 1$, cu $f = -1$, mai bună decît precedenta ; după un pas, concluzia că soluţia curentă este optimă. Ea corespunde vîrfului dublu încercuit. Remarcăm că s-au cercetat numai nouă dintre cele cinsprezece vîrfuri ale arborescenţei.

Să trecem acum la expunerea modificărilor necesare pentru ca algoritmul de mai sus să devină capabil să rezolve programe liniare

parţial întregi. Considerăm programul liniar parţial întreg: să se minimizeze

$$f(x_1, \ldots, x_n) = \underline{f}(x_1, \ldots, x_p) + \bar{f}(x_{p+1}, \ldots, x_n), \tag{6.71}$$

în condiţiile

$$f_i(x_1, \ldots, x_n) = \underline{f}_i(x_1, \ldots, x_p) + \bar{f}_i(x_{p+1}, \ldots, x_n) \leqslant 0, \tag{6.72}$$

$$(i = 1, \ldots, m),$$

$$0 \leqslant x_k \leqslant d_k, \quad (k = 1, \ldots, p), \tag{6.73}$$

$$x_k \geqslant 0, \ (k = p + 1, \ldots, n),$$

$$x_k \text{ întreg}, \ (k = 1, \ldots, p), \tag{6.74}$$

unde

$$\underline{f}(x_1, \ldots, x_p) = \sum_{k=1}^{p} c_k x_k, \ \bar{f}(x_{p+1}, \ldots, x_n) = \sum_{k=p+1}^{n} c_k x_k,$$

$$\underline{f}_i(x_1, \ldots, x_p) = \sum_{k=1}^{p} a_i^k x_k, \quad \bar{f}(x_{p+1}, \ldots, x_n) = \sum_{k=p+1}^{n} a_i^k x_k, \tag{6.75}$$

$$(i = 1, \ldots, m),$$

şi unde $1 \leqslant p < n$.

Pentru orice $r \leqslant p$ şi $\bar{x}_1, \ldots, \bar{x}_r$ numere fixate satisfăcînd condiţiile (6.73), (6.74) definim mulţimea

$$Q_r(\bar{x}_1, \ldots, \bar{x}_r) = \{X \mid x_k = \bar{x}_k, \ (k = 1, \ldots, r); \ 0 \leqslant x_k \leqslant d_k,$$

$$(k = 1, \ldots, p); \tag{6.76}$$

$$x_k \geqslant 0, \ (k = p + 1, \ldots, n); \ x_k \text{ întreg}, \ (k = 1, \ldots, p)\}.$$

Pe baza acestei definiţii vom asocia programului o arborescenţă cu $p + 1$ nivele, unde un vîrf oarecare de pe nivelul r, $(1 \leqslant r \leqslant p)$, va reprezenta o mulţime de vectori n-dimensionali $Q_r(\bar{x}_1, \ldots, \bar{x}_r)$. Cu alte cuvinte, arborescenţa tocmai definită este formată din pri-

Scanned with OKEN Scanner

mele $p+1$ nivele ale arborescenței definite în cazul total întreg, cu deosebirea esențială că orice mulțime de vectori asociată unui vîrf nu mai este finită; în particular, mulțimile asociate vîrfurilor de pe ultimul nivel nu mai conțin un singur vector.

Vom considera acum chestiunea marginilor asociate vîrfurilor, anticipînd faptul că separabilitatea variabilelor constrînse de cele neconstrînse, în funcția obiectiv și în restricții, va permite rezolvarea programului liniar parțial întreg printr-un algoritm care prezintă doar o modificare simplă față de algoritmul expus mai înainte pentru cazul total întreg. O propoziție simplă va permite formularea unor teoreme analoage teoremelor 6.3, 6.4.

Teorema 6.5. *Fiind date numerele* $\bar{x}_1, \ldots, \bar{x}_r$ *satisfăcînd condițiile* (6.73), (6.74) *pentru* $k = 1, \ldots, r$, $(r \leqslant p)$, *și funcția*

$$g(x_1, \ldots, x_n) = \underline{g}(x_1, \ldots, x_p) + \bar{g}(x_{p+1}, \ldots, x_n); \tag{6.77}$$

dacă există

$$\bar{\bar{g}} = \min \bar{g}(x_{p+1}, \ldots, x_n), \tag{6.78}$$

pentru valori nenegative ale variabilelor, atunci pe mulțimea $Q_r(\bar{x}_1, \ldots, \bar{x}_r)$ *definită prin* (6.76) *există*

$$g^r(\bar{x}_1, \ldots, \bar{x}_r) = \min g(\bar{x}_1, \ldots, \bar{x}_r, x_{r+1}, \ldots, x_n) \tag{6.79}$$

și

$$g^r(\bar{x}_1, \ldots, \bar{x}_r) = \min \underline{g}(\bar{x}_1, \ldots, \bar{x}_r, x_{r+1}, \ldots, x_p) + \bar{\bar{g}}. \tag{6.80}$$

Demonstrație. Rezultatul este o consecință evidentă a separabilității variabilelor constrînse de cele neconstrînse în funcția $g(x_1, \ldots, x_n)$.

Acum, putem demonstra două teoreme analoage teoremelor 6.3, 6.4.

Teorema 6.6. *Fie date numerele* $\bar{x}_1, \ldots, \bar{x}_r$ *satisfăcînd condițiile* (6.73), (6.74) *cu* $k = 1, \ldots, r$, $(r \leqslant p)$; *dacă pentru un indice* i *există*

$$\bar{\bar{f}}_i = \min \bar{f}_i(x_{p+1}, \ldots, x_n) \tag{6.81}$$

pentru valori nenegative ale variabilelor și

$$\min \underline{f}_i(\bar{x}_1, \ldots, \bar{x}_r, x_{r+1}, \ldots, x_p) + \bar{\bar{f}}_i > 0, \tag{6.82}$$

Scanned with OKEN Scanner

unde minimul este luat pentru valorile variabilelor $x_{r+1}, \ldots, x_p$ *satisfăcînd condițiile* (6.73), (6.74) *cu* $k = r + 1, \ldots, p$, *atunci în mulțimea* $Q_r(\bar{x}_1, \ldots, \bar{x}_r)$ *nu există nici o soluție admisibilă întreagă a programului parțial întreg.*

Demonstrație. Se aplică teorema 6.3, ținîndu-se seama de rezultatul din teorema 6.5 folosit pentru funcția $f_i(x_1, \ldots, x_n)$.

Teoroma 6.7. *Fie date numerele* $\bar{x}_1, \ldots, \bar{x}_r$ *satisfăcînd condițiile* (6.73), (6.74) *cu* $k = 1, \ldots, r$, $(r \leqslant p)$, *și numărul* S *o valoare disponibilă a funcției obiectiv* (6.71); *dacă există*

$$\bar{\bar{f}} = \min \bar{f}(x_{p+1}, \ldots, x_n) \tag{6.83}$$

pentru valori nenegative ale variabilelor și

$$\min f(\bar{x}_1, \ldots, \bar{x}_r, x_{r+1}, \ldots, x_p) + \bar{\bar{f}} \geqslant S, \tag{6.84}$$

unde minimul este luat pentru valorile variabilelor satisfăcînd condițiile (6.73), (6.74) *cu* $k = r+1, \ldots, p$, *atunci în mulțimea* $Q_r(\bar{x}_1, \ldots, \bar{x}_r)$ *nu există nici o soluție admisibilă întreagă „mai bună" decît cea disponibilă.*

Demonstrație. Se aplică teorema 6.4 ținîndu-se seamă de rezultatul din teorema 6.5 folosit pentru funcția $f(x_1, \ldots, x_n)$.

Remarcăm că teoremele 6.6, 6.7, care vor reprezenta criteriile de respingere ale algoritmului, sînt destul de slabe în comparație cu teoremele 6.3, 6.4, căci cer existența numerelor (6.81) și respectiv (6.83). Or aceste numere, în cazul unor funcții liniare, există (și sînt nule), dacă și numai dacă toți coeficienții variabilelor sînt nenegativi. Observăm de asemenea că teoremele nu au folosit liniaritatea funcțiilor considerate așa încît criteriile sînt aplicabile și pentru funcții neliniare; considerăm că acesta este un fapt important, deși nu-l vom folosi în continuare.

Întrucît orice vîrf de pe nivelul p nu mai reprezintă un singur vector, ci mulțimea vectorilor cu primele p componente fixate la anumite valori iar celelalte nenegative, nu mai dispunem de o sursă de determinare a unor soluții admisibile întregi. În cazul total întreg, cînd se calcula vectorul marginilor pentru un vîrf de pe nivelul n, prima componentă era valoarea funcției obiectiv pentru acest vector, iar celelalte permiteau stabilirea admisibilității vectorului; prin compararea valorii funcției obiectiv cu valoarea cu-

Scanned with OKEN Scanner

rentă stabileam în caz de admisibilitate dacă vectorul în chestiune reprezintă o soluţie admisibilă întreagă „mai bună" decît cea curentă, iar în caz afirmativ o reţineam. Prin urmare, în cazul parţial întreg mai este necesară o cale de determinare a soluţiilor admisibile întregi, căci vectorul marginilor nu poate fi folosit pentru aceasta; el verifică numai îndeplinirea unor condiţii suficiente de neadmisibilitate şi neameliorare. Din acest motiv, dacă s-ar cunoaşte dinainte că rezultatul folosirii lui nu aduce respingerea mulţimii de vectori reprezentată printr-un vîrf de pe nivelul p, atunci el nici nu ar mai trebui calculat. Totuşi, calculul este necesar, căci ne poate scuti de efortul de calcul pentru aflarea unei soluţii admisibile întregi a programului liniar parţial întreg.

Pentru determinarea unei soluţii admisibile întregi dintr-o mulţime $Q_p(\bar{x}_1, \ldots, \bar{x}_p)$ corespunzătoare unui vîrf al arborescenţei situat pe nivelul p, ţinîndu-se seamă de faptul că dorim să obţinem o soluţie cît mai apropiată de o soluţie optimă a programului liniar parţial întreg, vom rezolva un program liniar obişnuit: să se minimizeze

$$\bar{f}(x_{p+1}, \ldots, x_n) \tag{6.85}$$

în condiţiile

$$\begin{gathered}\bar{f}_i(x_{p+1}, \ldots, x_n) \leqslant -\underline{f}_i(\bar{x}_1, \ldots, \bar{x}_p), \ (i = 1, \ldots, m), \\ x_k \geqslant 0, \ (k = p+1, \ldots, n).\end{gathered} \tag{6.86}$$

Dacă acest program liniar dă o soluţie optimă şi valoarea funcţiei obiectiv corespunzătoare f^* este mai mică decît S, vom schimba soluţia curentă şi valoarea curentă a lui S, prin soluţia optimă găsită completată cu valorile fixate ale primelor p variabile şi numărul f^*; dacă soluţia optimă găsită dă o valoare $f^* \geqslant S$, sau programul liniar nu are soluţii admisibile, ne vom întoarce la un vîrf precedent al arborescenţei. O situaţie nouă poate apărea aici, deosebită de cele întîlnite în aplicarea algoritmului anterior: dacă programul liniar are funcţia obiectiv nemărginită inferior pe mulţimea soluţiilor admisibile, atunci programul liniar parţial întreg are funcţia obiectiv nemărginită inferior pe mulţimea soluţiilor admisibile întregi, şi astfel cercetarea vîrfurilor arborescenţei trebuie oprită, cu toate că acestea nu au fost numaidecît epuizate.

Scanned with OKEN Scanner

Cu precizările făcute asupra calculului vectorului marginilor pentru un vîrf oarecare și asupra cercetării vîrfurilor de pe ultimul nivel al arborescenței, am relevat deosebirile dintre algoritmul pentru rezolvarea cazului parțial întreg și algoritmul pentru rezolvarea cazului total întreg. De aceea, considerăm că nu mai este necesar să formulăm noul algoritm, căci este utilizabilă formularea algoritmului precedent cu modificarea punctului b_1) înlocuit prin următorul:

b_1') dacă vîrful cercetat este pe ultimul nivel, rezolvăm programul liniar obișnuit (6.85), (6.86) și acționăm astfel:

— schimbăm soluția curentă și numărul S, cînd există o soluție optimă și $f^* < S$, apoi procedăm ca la a_1) sau a_2), după caz;

— nu schimbăm soluția curentă și numărul S, cînd programul nu are soluții admisibile sau are o soluție optimă dar $f^* \geqslant S$, apoi procedăm ca la a_1) sau a_2), după caz;

— oprim algoritmul cînd programul liniar (6.85), (6.86) are funcția obiectiv nemărginită inferior.

Exemplul 6.8. Să considerăm un program liniar parțial întreg: să se minimizeze

$$f = x_1 + x_2 - 2x_3, \tag{6.87}$$

în condițiile

$$f_1 = 2x_1 - x_2 + x_3 \leqslant 0, \quad x_1, x_2 \in \{0, 1\}, \; x_3 \geqslant 0. \tag{6.88}$$

Remarcăm de la început că nu există minimul părții din f care conține variabila neconstrînsă x_3, datorită prezenței coeficientului negativ; de aceea, nu vom putea folosi criteriul de respingere relativ la neameliorarea valorii funcției obiectiv. Există totuși minimul părții din f_1 care conține variabila x_3 și $\bar{f}_1 = 0$, așa încît vom putea folosi celălalt criteriu de respingere. Arborescența asociată programului are trei nivele.

Se ia inițial $S = \max f = 2$ și un vector arbitrar drept soluție curentă. Pașii algoritmului ne conduc la considerarea părții arborescenței asociate programului, date în fig. 3

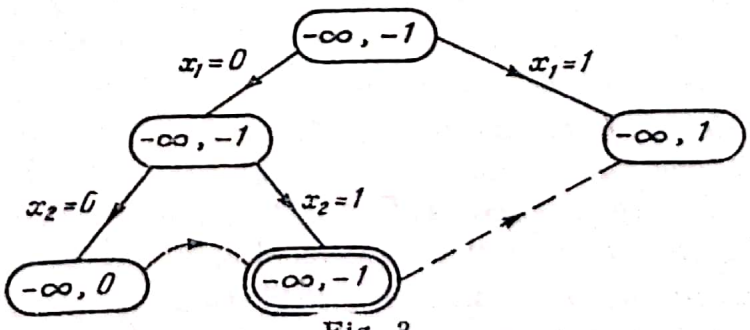


Fig. 3

Perechile de numere scrise în interiorul vîrfurilor reprezintă valorile componentelor vectorului marginilor; simbolul $-\infty$ înseamnă imposibilitatea utilizării criteriului bazat pe valoarea componentei respective. Arcele cu linie punctată indică mersul înapoi. După trei paşi s-a obţinut un vîrf de pe ultimul nivel nerespins de criteriu, s-a rezolvat programul liniar obişnuit următor: să se minimizeze $-2x_3$ cu restricţiile $x_3 \leqslant 0$, $x_3 \geqslant 0$, şi s-a găsit o soluţie optimă $x_3 = 0$, care dă o soluţie admisibilă întreagă $x_1 = 0$, $x_2 = 0$, $x_3 = 0$ cu $f^* = 0$. Cum $f^* < S$, s-a luat aceasta drept soluţie curentă şi $S = 0$. În pasul următor s-a întîlnit aceeaşi situaţie; s-a rezolvat programul liniar obişnuit următor: să se minimizeze $-2x_3$ cu restricţiile $x_3 \leqslant 1$, $x_3 \geqslant 0$, şi s-a găsit o soluţie optimă $x_3 = 1$, care dă o soluţie admisibilă întreagă $x_1 = 0$, $x_2 = 1$, $x_3 = 1$ cu $f^* = 1$. Cum $f^* < S$, s-a luat aceasta drept soluţie curentă şi $S = 1$. În pasul următor s-a obţinut concluzia că soluţia curentă este o soluţie optimă a programului liniar parţial întreg; ea corespunde vîrfului dublu încercuit. Remarcăm că s-au cercetat numai cinci dintre cele şapte vîrfuri ale arborescenţei.

Scanned with OKEN Scanner

7

PROBLEMA TRANSPORTURILOR

7.1. REȚEA DE TRANSPORT, FLUX, SECȚIUNE, PROBLEMA FLUXULUI MAXIM, PROBLEMA SECȚIUNII MINIME

Problema de transport, așa cum a fost formulată în § 2.1.3, a sugerat numeroase extensiuni bazate mai ales pe încercările de a adapta modelul la diferitele situații reale. Din punct de vedere matematic, aceste extensiuni au fost dezvoltate prin elaborarea unor noi capitole în disciplina matematică denumită teoria grafurilor. Uneori, se consideră în literatură că întreg ansamblul de rezultate legate de problema de transport constituie o nouă disciplină matematică denumită teoria rețelelor de transport. Prima monografie consacrată acestui grup de probleme este datorată lui L. R. Ford și D.R. Fulkerson [13]. În continuare vom schița conținutul teoriei rețelelor de transport, arătînd la momentul potrivit locul problemei de transport din programarea liniară. Oricum, cele mai simple probleme ale teoriei sînt tot probleme de programare liniară, dar aspectul lor matematic mai simplu a permis elaborarea unei serii de metode proprii de rezolvare.

7.1.1. *REȚEA DE TRANSPORT*

Un graf finit orientat $G = (V, U)$ este cuplul format dintr-o mulțime finită V, denumită mulțimea vîrfurilor, și o mulțime de perechi ordonate de elemente din V, notată cu U, denumită mulțimea arcelor (v. § 6.4). Graful este un obiect matematic care poate reprezenta teoretic foarte multe obiecte concrete, cum ar fi, de exemplu : rețeaua străzilor dintr-un oraș, rețeaua electrică a unei localități, rețeaua nervoasă din corpul uman etc. În cele ce urmează vom privi totdeauna un graf ca o rețea pe care se transportă ceva : un produs, curent electric, influx nervos, etc. ; de aici rezultă necesitatea de a defini inițial conceptul matematic de rețea.

Fie $G = (V, U)$ un graf finit orientat pe care sînt definite una sau mai multe funcții. Desigur, în fiecare problemă apar anumite

Scanned with OKEN Scanner

funcţii definite pe graf, dar totuşi putem prezenta pe cele mai des folosite în problemele teoriei reţelelor de transport.

Funcţia de cerere este o funcţie definită pe mulţimea vîrfurilor V, prin care se asociază fiecărui vîrf v un număr, notat $d(v)$ şi denumit cererea vîrfului v. Funcţia de cerere împarte mulţimea vîrfurilor în trei submulţimi:

$$D = \{v \mid v \in V,\ d(v) < 0\},$$

$$T = \{v \mid v \in V,\ d(v) = 0\}, \tag{7.1}$$

$$C = \{v \mid v \in V,\ d(v) > 0\},$$

dintre care T ar putea fi şi vidă. Cum un vîrf cu cerere negativă este interpretat ca vîrf cu un disponibil egal cu $|d(v)|$, vîrfurile grafului vor fi denumite astfel: $v \in D$ este denumit punct de disponibil (sursă), $v \in T$ este denumit punct de tranzit, şi $v \in C$ este denumit punct de cerere (destinaţie). Este clar că denumirile introduse provin din modelele matematice ale unor probleme economice şi că există şi altele provenite din modelele matematice ale altor tipuri de probleme. Dintre acestea vom folosi uneori acele de intrare = punct de disponibil şi de ieşire = punct de cerere.

Funcţia de capacitate este o funcţie definită pe mulţimea arcelor U, prin care se asociază fiecărui arc $u \in U$ un număr nenegativ, notat $k(u)$ şi denumit capacitatea arcului u. Denumirea însăşi precizează că numărul $k(u)$ este interpretat de obicei ca o barieră superioară pentru o cantitate ce urmează a fi transportată în lungul arcului. Se înţelege, există probleme unde trebuie să se asigure o anumită cantitate minimală transportată în lungul fiecărui arc, adică probleme care necesită introducerea unei funcţii de capacitate inferioară; din punct de vedere matematic prezenţa în problemă a unei asemenea funcţii nu introduce dificultăţi noi, aşa încît presupunem de obicei că există o singură funcţie de capacitate.

Funcţia de cost este o funcţie definită pe mulţimea arcelor U, prin care se asociază fiecărui arc $u \in U$ un număr nenegativ, notat $c(u)$ şi denumit costul arcului u. Denumirea precizează de asemenea că numărul $c(u)$ este interpretat adesea ca un cost al transportului unei unităţi în lungul arcului, indiferent dacă acest număr este efectiv un cost, sau un timp de parcurs, sau are un alt sens practic.

Scanned with OKEN Scanner

Pentru funcţiile definite pe mulţimea vîrfurilor se poate da o reprezentare sub formă de vector : dacă V este o mulţime ordonată, cu $|V| = n$ şi elementele v_i, $(i = 1, \ldots, n)$, putem reprezenta funcţia de cerere $d(v)$ sub forma unui vector $d = (d_i)$. Pentru funcţiile definite pe mulţimea arcelor se poate da o reprezentare sub formă de matrice, deşi aici se prezintă de cele mai multe ori inconvenientul că mulţimea U nu conţine toate perechile ordonate de elemente din V. Dacă nu se prezintă acest ultim caz, atunci este clar că ordinea din mulţimea V ataşează fiecărei perechi ordonate de elemente, adică unui arc, o pereche de indici ce se transmit şi la valoarea funcţiei corespunzătoare arcului. Dacă în U nu sînt anumite perechi ordonate de elemente din V şi, prin urmare, nu sînt definite valorile funcţiei pentru aceste arce, este de obicei uşor să considerăm că arcele în chestiune s-au adăugat şi să definim valori convenabile ale funcţiei. De exemplu, dacă în U nu există un arc corespunzător unei anumite perechi de vîrfuri şi discutăm despre funcţia de capacitate, putem introduce arcul în U şi totodată atribui capacităţii lui o valoare nulă ; pe această cale, putem defini o matrice a capacităţilor $K = (k_{ij})$, $(i, j = 1, \ldots, n)$. Pentru funcţia de cost se poate de obicei proceda asemănător, numai că valoarea costului atribuită unui arc practic inexistent se ia adesea un număr foarte mare pozitiv ; aceasta se explică prin considerarea curentă a unor probleme de minimizare a costului, unde cu asemenea valori ale costurilor se va obţine excluderea din soluţie a arcelor inexistente. Pentru funcţiile definite pe mulţimea arcelor se poate da şi o reprezentare vectorială, dacă se ordonează mulţimea arcelor, dar în continuare vom adopta de obicei o reprezentare matriceală.

Se numeşte *reţea de transport* un obiect matematic format dintr-un graf şi mai multe funcţii definite pe graf.

Am discutat mai sus despre reprezentarea numerică a funcţiilor definite pe graful aparţinînd unei reţele de transport. Se înţelege, graful însuşi trebuie reprezentat numeric în vederea rezolvării problemelor cu ajutorul calculatorului. Un graf poate fi reprezentat în mai multe moduri, dintre care alegem aici numai reprezentarea prin matricea asociată şi prin matricea de incidenţă. Se numeşte *matrice asociată* grafului $G = (V, U)$, cu $|V| = n$, o matrice de ordinul n, notată $M = (m_{ij})$, definită astfel

$$m_{ij} = \begin{cases} 1, & \text{dacă } (v_i, v_j) \in U, \\ 0, & \text{dacă } (v_i, v_j) \notin U, \end{cases} \qquad (i, j = 1, \ldots, n). \qquad (7.2)$$

Aşadar, un graf este reprezentabil în mod univoc printr-o matrice pătratică cu elemente zero şi unu. Se numeşte *matrice de incidenţă* a grafului $G = (V, U)$ cu $|V| = n$, $|U| = m$, o matrice dreptunghiulară de tip $n \times m$, notată $A = (a_{ik})$, definită după cum urmează:

$$a_{ik} = \begin{cases} 1, & \text{dacă } u_k = (x_i, x_j),\ u_k \in U, \\ -1, & \text{dacă } u_k = (x_j, x_i),\ u_k \in U, \\ 0 & \text{în celelalte cazuri} \end{cases} \tag{7.3}$$

$$(i = 1, \ldots, n;\ k = 1, \ldots, m).$$

Cu alte cuvinte, $a_{ik} = 1$ cînd arcul u_k iese din vîrful v_i, $a_{ik} = -1$ cînd arcul u_k intră în vîrful v_i, $a_{ik} = 0$ cînd arcul u_k nu are vîrful v_i drept extremitate, nici iniţială, nici finală. Remarcăm că o asemenea reprezentare admit numai grafurile care nu conţin arce cu extremităţi identice, arce denumite bucle; într-adevăr, un asemenea arc iese dintr-un anumit vîrf şi intră în acelaşi vîrf, aşa încît elementul corespunzător vîrfului şi buclei nu este definit. Matricea de incidenţă are proprietatea evidentă că orice coloană a sa conţine numai două elemente nenule, unul egal cu 1 şi unul egal cu -1.

În literatură matricea asociată unui graf mai este denumită matricea vîrfuri-vîrfuri, în timp ce matricea de incidenţă este denumită matricea vîrfuri-arce. Aceste denumiri precizează că în prima matrice fiecare linie şi fiecare coloană corespunde cîte unui vîrf, în timp ce în a doua fiecare linie corespunde unui vîrf iar fiecare coloană unui arc. Pentru rezolvarea unor tipuri de probleme este convenabilă matricea asociată, pentru rezolvarea altora este convenabilă matricea de incidenţă, iar uneori este mai bine ca structura grafului să fie reprezentată prin alte matrice ataşate grafului.

Exemplul 7.1. Graful cu patru vîrfuri şi şase arce definit prin: $V = \{a, b, c, d\}$ şi $U = \{(a, b), (b, c), (b, d), (a, d), (d, c), (c, c)\}$ are matricea asociată

$$M = \begin{pmatrix} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{7.4}$$

Scanned with OKEN Scanner

Observăm că există o buclă, (c, c), deci nu putem scrie matricea de incidență definită mai sus. Să scriem matricea de incidență pentru graful cu patru vîrfuri și cinci arce definit prin: $V^* = V$, $U^* = U - \{(c\ c)\}$, presupunînd că ordinea de pe mulțimea arcelor este cea definită de enumerarea elementelor lui U:

$$A^* = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 \\ -1 & 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & -1 \\ 0 & 0 & -1 & -1 & 1 \end{pmatrix}. \tag{7.5}$$

7.1.2. *FLUX SECȚIUNE*

Considerăm mai întîi un tip particular de rețea, rețeaua redusă, deoarece rezolvarea unor probleme de flux în rețele mai generale se realizează prin utilizarea unei rețele reduse asociate.

Se numește *rețea redusă* o rețea ale cărei elemente au următoarele proprietăți:

— graful $G = (V, U)$ posedă un vîrf, fie v_1, în care nu intră nici un arc, și un vîrf, fie v_n, din care nu iese nici un arc;

— funcția de cerere este definită prin

$$d(v_i) = \begin{cases} -v, & i = 1, \\ 0, & i \neq 1, n, \\ v, & i = n, \end{cases} \tag{7.6}$$

unde v este un număr nenegativ;

— funcția de capacitate este definită printr-o matrice $K = (k_{ij})$ cu elemente nenegative; se pun elemente nule pentru perechile ordonate de vîrfuri ce nu constituie extremități de arce ale grafului.

După cum se vede, în rețeaua redusă de care ne ocupăm nu este definită o funcție de cost; mai mult, chiar numărul v din definiția funcției de cerere va fi considerat în cele ce urmează ca valoare a unui parametru.

Se numește *flux* într-o rețea redusă dată o funcție definită pe mulțimea arcelor care satisface trei grupuri de condiții: condiții de conservare, condiții de capacitate și condiții de nenegativitate.

Scanned with OKEN Scanner

Dacă se consideră fluxul definit matriceal, $F = (f_{ij})$, cele trei grupuri de condiţii din definiţie sînt următoarele:

$$\sum_{j=1}^{n} f_{ij} - \sum_{j=1}^{n} f_{ji} = \begin{cases} v, & i = 1, \\ 0, & i \neq 1, n, \\ -v, & i = n, \end{cases} \tag{7.7}$$

$$f_{ij} \leqslant k_{ij}, \quad (i, j = 1, \ldots, n), \tag{7.8}$$

$$f_{ij} \geqslant 0, \quad (i, j = 1, \ldots, n). \tag{7.9}$$

Condiţiile de conservare exprimă faptul că pentru orice vîrf al reţelei, diferit de intrare şi ieşire, suma fluxurilor pe arcele ce intră în vîrf este egală cu suma fluxurilor pe arcele ce ies din vîrf; cum în x_1 nu intră nici un arc şi, conform condiţiilor (7.8), avem $f_{i1} = 0$, $(i = 1, \ldots, n)$, suma fluxurilor ce ies din intrare, adică intră în reţea, este egală cu v; cum din x_n nu iese nici un arc şi, conform condiţiilor (7.8), avem $f_{ni} = 0$, $(i = 1, \ldots, n)$, suma fluxurilor ce intră în ieşire, adică ies din reţea, este egală tot cu v. Condiţiile (7.8) exprimă faptul că pe orice arc fluxul nu poate depăşi capacitatea; ca urmare, pe orice arc de capacitate nulă, ţinînd seamă şi de condiţiile (7.9), fluxul este nul. Condiţiile (7.9) exprimă sensul practic al fluxului pe arc, f_{ij} fiind o anumită cantitate de produs transportat de la vîrful v_i la vîrful v_j.

Dacă mulţimea arcelor este ordonată şi, ca urmare, funcţia de capacitate este dată printr-un vector, fluxul φ se defineşte şi el ca un vector, satisfăcînd cele trei grupuri de condiţii menţionate, condiţii care se exprimă analitic astfel:

$$\sum_{u_h \in U_i^+} \varphi_h - \sum_{u_h \in U_i^-} \varphi_h = \begin{cases} v, & i = 1, \\ 0, & i \neq 1, n, \\ -v, & i = n. \end{cases} \tag{7.10}$$

$$\begin{aligned} &\varphi_h \leqslant k_h, \quad (h = 1, \ldots, m), \\ &\varphi_h \geqslant 0, \quad (h = 1, \ldots, m), \end{aligned} \tag{7.11}$$

unde

$$U_i^+ = \{u_h \mid u_h = (x_i, x_j),\ h \in \{1, \ldots, m\}\},$$
$$U_i^- = \{u_h \mid u_h = (x_j, x_i),\ h \in \{1, \ldots, m\}\}, \tag{7.12}$$
$$(i = 1, \ldots, n).$$

Aceste condiţii se scriu matriceal

$$A\varphi = b, \quad 0 \leqslant \varphi \leqslant k, \tag{7.13}$$

unde A este matricea de incidenţă a grafului, b este un vector n-dimensional cu componentele $b_i = 0$, $(i \neq 1, n)$, $b_1 = v$, $b_n = -v$, apoi k este vectorul capacitate, iar φ este vectorul flux.

Exemplul 7.2. Considerăm reţeaua formată din graful $G = (V^*, U^*)$ dat în exemplul 7.1, funcţia de cerere definită prin numărul $v = 6$ şi funcţia de capacitate definită prin vectorul $k = (4, 2, 2, 3, 4)'$. Un flux în această reţea este definit prin vectorul $\varphi = (3, 2, 1, 3, 4)'$. Faptul că vectorul φ reprezintă un flux în reţea se vede uşor în fig. 4; numerele scrise pe fiecare arc sînt capacităţile arcelor, iar în paranteze fluxurile.

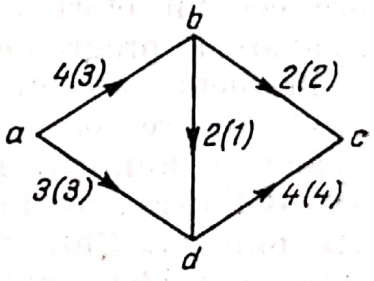


Fig 4.

Fie H o mulţime de vîrfuri astfel ca $x_1 \in H$, $x_n \notin H$. Mulţimea arcelor cu extremitatea iniţială în H şi extremitatea finală în afara lui H se numeşte *secţiunea reţelei* determinată de mulţimea de vîrfuri H. Se numeşte *capacitatea secţiunii* determinată de mulţimea de vîrfuri H suma capacităţilor arcelor secţiunii. În continuare vom nota secţiunea determinată de mulţimea H cu S_H, iar capacitatea sa cu $k(S_H)$.

Exemplul 7.3. Reţeaua considerată în exemplul 7.2 are mai multe secţiuni. Una dintre ele este determinată, de exemplu, de mulţimea $H = \{a, b\}$ şi avem $S_H = \{(b, c), (b, d), (a, d)\}$, cu $k(S_H) = 7$.

Între mulţimea fluxurilor corespunzătoare unui v dat şi mulţimea secţiunilor reţelei, deşi există o legătură importantă, prezentată în continuare, există şi o deosebire esenţială: în timp ce mulţimea fluxurilor este în general infinită (poate fi şi vidă), mulţimea secţiunilor este totdeauna finită. Acest fapt are drept consecinţă acela că două probleme fundamentale ale teoriei reţelelor de trans-

Scanned with OKEN Scanner

port, problema fluxului maxim şi problema secţiunii minime, sînt două probleme matematice de natură diferită.

7.1.3. *PROBLEMA FLUXULUI MAXIM, PROBLEMA SECŢIUNII MINIME*

Problema fluxului maxim se formulează într-o reţea redusă în care valoarea funcţiei de cerere în intrarea x_1, adică $-v$, este privită ca parametru. Problema este următoarea : să se maximizeze funcţia $f = v$ în condiţiile (7.7) — (7.9), sau sub forma matriceală (7.13). Aşadar, în problemă variabile sînt considerate valorile fluxului pe arce şi v. Cum v reprezintă suma valorilor fluxului pe arcele care ies din x_1, sau suma valorilor fluxului pe arcele care intră în x_n, valoarea lui v corespunzătoare unui anumit flux în reţea se numeşte *valoarea fluxului*. De aceea, problema tocmai formulată se numeşte *problema fluxului maxim*. Remarcăm că problema este un program liniar a cărui matrice are un aspect particular. De aceea, pentru rezolvarea problemei există metode specifice de rezolvare care cer un efort de calcul mai mic decît metodele generale de rezolvare a programelor liniare, expuse în cap. 3.

Problema secţiunii minime într-o reţea redusă se formulează astfel : dintre toate secţiunile reţelei să se determine secţiunea de capacitate minimă. Este clar că denumirea problemei se justifică prin definiţia : se numeşte *secţiune minimă* secţiunea de capacitate minimă. Cum mulţimea secţiunilor este finită problema este numai o problemă numerică ; numărul secţiunilor poate fi foarte mare şi cercetarea întregii mulţimi ar cere un efort de calcul foarte mare.

Anticipăm faptul că rezolvarea problemei secţiunii minime se face în acelaşi timp cu rezolvarea problemei fluxului maxim, cu ajutorul aceluiaşi algoritm. Problema secţiunii minime se poate dealtfel formula ca program liniar, dar un program liniar în numere întregi.

7.2. ALGORITMUL LUI L. R. FORD ŞI D. R. FULKERSON PENTRU REZOLVAREA PROBLEMEI FLUXULUI MAXIM ŞI A PROBLEMEI SECŢIUNII MINIME

Cel mai cunoscut algoritm pentru rezolvarea problemei fluxului maxim într-o reţea redusă este algoritmul de marcaj al lui L. R. Ford şi D. R. Fulkerson, (v. [13], § I.5). Expunem algoritmul

Scanned with OKEN Scanner

împreună cu rezultatele care-l justifică, stabilind şi cîteva rezultate preliminare.

Considerăm o reţea redusă R şi întrebuinţăm notaţii matriceale. Problema fluxului maxim se formulează astfel: să se maximizeze v în condiţiile

$$\sum_{j=1}^{n} f_{ij} - \sum_{j=1}^{n} f_{ji} = \begin{cases} v, & i = 1, \\ 0, & i \neq 1, n, \\ -v, & i = n, \end{cases} \tag{7.14}$$

$$(i = 1, \ldots, n),$$

şi

$$0 \leqslant f_{ij} \leqslant k_{ij}, \ (i, j = 1, \ldots, n). \tag{7.15}$$

Teorema 7.1. *Oricare ar fi fluxul $F = (f_{ij})$ cu valoarea corespunzătoare v_F şi secţiunea S_H cu capacitatea $k(S_H)$, în reţeaua redusă R, avem*

$$v_F \leqslant k(S_H). \tag{7.16}$$

Demonstraţie. Dacă adunăm ecuaţiile (7.14) corespunzătoare vîrfurilor $v_i \in H$ şi ţinem seamă atît de reducerile de termeni din membrul stîng cît şi de faptul că $v_1 \in H$, $v_n \notin H$, rezultă egalitatea

$$\sum_{i \in H,\, j \notin H} f_{ij} - \sum_{i \in H,\, j \notin H} f_{ij} = v_F; \tag{7.17}$$

fiecare termen din prima sumă este mai mic sau egal cu capacitatea arcului corespunzător, iar fiecare termen din a doua sumă este nenegativ, conform inegalităţilor (7.15), astfel încît inegalitatea (7.16) rezultă din (7.17).

Teorema 7.2. *Dacă $F = (f_{ij})$ este un flux cu valoarea v_F şi S_H este o secţiune cu capacitatea $k(S_H)$ şi $v_F = k(S_H)$, atunci F este un flux maxim şi S_H este o secţiune minimă, în reţeaua redusă R.*

Demonstraţie. Într-adevăr, dacă ar exista un flux F^* cu $v_{F^*} > v_F$, atunci cum $v_F = k(S_H)$ ar exista un flux F^* şi o secţiune S_H astfel ca $v_{F^*} > k(S_H)$, contrar teoremei 7.1. La fel se demonstrează că S_H este o secţiune minimă.

Scanned with OKEN Scanner

Problema ce se pune este dacă poate exista un flux maxim F cu valoarea v_F și o secțiune minimă S_H cu capacitatea $k(S_H)$, astfel ca $v_F < k(S_H)$. Răspunsul este negativ, cum o arată teorema următoare:

Teorema 7.3. *Pentru orice cuplu format dintr-un flux maximal F cu valoarea v_F și o secțiune minimală S_H cu capacitatea $k(S_H)$, într-o rețea redusă R, avem $v_F = k(S_H)$.*

Demonstrația teoremei 7.3 este constructivă; de aceea să prezentăm în prealabil algoritmul lui L. R. Ford și D. R. Fulkerson pentru rezolvarea problemei fluxului maxim și problemei secțiunii minime, algoritm care reprezintă o particularizare a procedeului din această demonstrație.

Ideea centrală a algoritmului este de a se determina un lanț de la v_1 la v_N, satisfăcînd anumite condiții, astfel încît modificînd un flux deja cunoscut să obținem un flux cu valoare mai mare. Prin lanț în rețea de la v_1 la v_N se înțelege o succesiune alternativă de vîrfuri și arce, în care orice arc are drept extremități vîrful ce-l precede și vîrful ce-l succede, sau invers. Dacă funcția de capacitate are numai valori întregi, algoritmul este convergent, adică rezolvă cele două probleme după un număr finit de pași.

Observăm că $v = 0$, $f_{ij} = 0$, $(i, j = 1, \ldots, n)$, este un sistem de valori ale variabilelor care satisface condițiile (7.14), (7.15); vom denumi acest flux cu valoarea zero *fluxul nul*.

Să presupunem că funcția de capacitate are numai valori întregi nenegative. Să mai presupunem că, după r pași ai algoritmului, $r \geqslant 0$, este cunoscut un flux întreg $F = (f_{ij})$ cu valoarea v_F; dacă $r = 0$ și nu este cunoscut vreun alt flux întreg, se poate considera ca flux de plecare fluxul nul. Atunci, pasul $r + 1$, al algoritmului constă în două operații principale: a) determinarea lanțului pe care se va modifica fluxul și (dacă este cazul), b) modificarea fluxului. Să ne ocupăm pe rînd de cele două operații:

a) Se caută un lanț μ de la v_1 la v_n cu proprietatea că

$$(v_i, v_j) \in \mu^+ \Rightarrow f_{ij} < k_{ij}, \tag{7.18}$$

$$(v_j, v_i) \in \mu^- \Rightarrow f_{ji} > 0, \tag{7.19}$$

unde μ^+ și μ^- reprezintă mulțimile de arce din μ parcurse în sens direct și respectiv invers, cînd μ este parcurs de la v_1 la v_n. Deter-

minarea unui lanț μ cu proprietățile (7.18), (7.19) se realizează printr-un procedeu de marcare a vîrfurilor rețelei. Prin aplicarea procedeului se atribuie fiecărui vîrf ce poate fi marcat două mărci: prima care reprezintă vîrful de la care a fost marcat vîrful considerat, însoțită de semnul + sau —, după cum arcul ce a permis marcarea a fost parcurs pentru aceasta în sens direct sau invers, a doua care reprezintă cantitatea maximă de flux suplimentar ce poate sosi în vîrful considerat prin modificarea fluxului pe arcele lanțului de la v_1 la acel vîrf, determinat de procedeul de marcaj.

Procedeul de marcare constă în aplicarea următoarelor reguli de marcare:

— vîrfului v_1 i se atribuie numai marca a doua: un număr pozitiv oricît de mare;

— dacă v_i este un vîrf marcat cu $(\pm z,\ \varepsilon(v_i))$ și există un arc $(v_i,\ v_j)$ în rețea, cu v_j încă nemarcat și cu $f_{ij} < k_{ij}$, vom marca v_j cu $(+v_i,\ \varepsilon(v_j))$, unde

$$\varepsilon(v_j) = \min(\varepsilon(v_i),\ k_{ij} - f_{ij}); \tag{7.20}$$

— dacă v_i este un vîrf marcat cu $(\pm z,\ \varepsilon(v_i))$ și există un arc $(v_j,\ v_i)$ în rețea, cu v_j încă nemarcat și $f_{ji} > 0$, vom marca v_j cu $(-v_i,\ \varepsilon(v_j))$, unde

$$\varepsilon(v_j) = \min\ (\varepsilon(v_i),\ f_{ji}). \tag{7.21}$$

Observăm că dacă fluxul inițial era întreg și funcția de capacitate este o funcție cu valori întregi nenegative, toate mărcile numerice $\varepsilon(v_i)$ atribuite sînt numere întregi pozitive.

Aplicarea procedeului de marcare se încheie cu una dintre următoarele două situații: a_1) procedeul se oprește, datorită imposibilității de a mai atribui mărci noi, și vîrful v_n nu a fost marcat; a_2) vîrful v_n tocmai a fost marcat. Vom demonstra că în primul caz problemele sînt rezolvate și în al doilea caz putem construi un flux întreg cu valoare mai mare decît fluxul disponibil.

C a z u l a_1). Fie H mulțimea de vîrfuri care au fost deja marcate în cursul aplicării procedeului. Avem $v_1 \in H$ și $v_n \notin H$, așa că H determină o secțiune S_H în rețea. Pe arcele $(v_i,\ v_j)$ ale secțiunii avem $f_{ij} = k_{ij}$, altfel v_j ar fi putut fi marcat; pe arcele $(v_j,\ v_i) \in U$, $v_i \in H$, $v_j \notin H$, avem $f_{ji} = 0$, altfel v_j ar fi putut fi marcat. Dacă

ţinem seamă de egalitatea (7.17), valabilă pentru orice flux şi secţiune în reţea, şi de implicaţiile

$$\begin{aligned}(v_i, v_j) \in U,\ v_i \in H,\ v_j \notin H &\Rightarrow f_{ij} = k_{ij},\\ (v_j, v_i) \in U,\ v_i \in H,\ v_j \notin H &\Rightarrow f_{ji} = 0,\end{aligned} \tag{7.22}$$

obţinem $k(S_H) = v_F$; prin urmare, conform teoremei 7.2 fluxul întreg disponibil este maxim iar secţiunea determinată de procedeul de marcaj este minimă. Aşadar, ambele probleme sînt rezolvate.

C a z u l a$_2$). În acest caz este determinat un lanţ $\mu = [v_1, v_{i_1}, \ldots, v_{i_p}, v_n]$ cu proprietăţile (7.18), (7.19). Într-adevăr, construind lanţul de la v_n la v_1, anume luînd drept vîrf ce precedă în lanţ un vîrf dat tocmai acela de la care a fost marcat, arcele lanţului au proprietăţile (7.18), (7.19), conform regulilor de marcaj. Mai mult, dacă $\varepsilon(v_n) = \varepsilon$ este a doua marcă a vîrfului v_n, atunci

$$\begin{aligned}(v_i, v_j) \in \mu^+ &\Rightarrow k_{ij} - f_{ij} \geqslant \varepsilon,\\ (v_j, v_i) \in \mu^- &\Rightarrow f_{ji} \geqslant \varepsilon,\end{aligned} \tag{7.23}$$

şi există cel puţin un arc al lanţului astfel ca în inegalitatea (7.23) corespunzătoare să avem egalitate. Scopul operaţiei a) fiind atins se trece la operaţia b).

b) Se construieşte un nou flux $F' = (f'_{ij})$ cu valoarea $V_{F'}$, prin modificarea fluxului disponibil conform formulei

$$f'_{ij} = \begin{cases} f_{ij} + \varepsilon, & (v_i, v_j) \in \mu^+,\\ f_{ij} - \varepsilon, & (v_i, v_j) \in \mu^-,\\ f_{ij}, & (v_i, v_j) \notin \mu,\end{cases} \tag{7.24}$$

$$(i, j = 1, \ldots, n).$$

Se demonstrează uşor că formula (7.24) defineşte un flux întreg, dacă ţinem seamă şi de condiţiile (7.23); valoarea noului flux, este $v_{F'} = v_F + \varepsilon$. Urmează să trecem la un nou pas al algoritmului, folosind fluxul întreg F' ca flux disponibil.

Deoarece şirul valorilor fluxurilor întregi obţinute în paşii algoritmului este un şir strict crescător de numere întregi, care este

și mărginit superior de capacitatea secțiunii minime, rezultă că după un număr finit de pași algoritmul va fi oprit în cursul operației a) a unui pas, unde se va prezenta cazul a_1). Astfel, convergența algoritmului este asigurată.

Remarcăm că algoritmul s-ar putea să nu fie convergent dacă renunțăm la ipoteza integrității valorilor funcției de capacitate. Se înțelege, dacă aceste valori sînt raționale, încă este asigurată convergența algoritmului. Remarcăm de asemenea că există variante ale algoritmului, în care determinarea lanțului necesar în fiece pas se realizează după reguli suplimentare capabile să accelereze convergența.

Exemplul 7.3. Să considerăm rețeaua din fig. 5, pe arcele căreia sînt date valorile funcției de capacitate și, în paranteze fluxurile disponibile corespunzătoare. Sînt date de asemenea mărcile vîrfurilor atribuite ca urmare a operației a) din noul pas al algoritmului. Operația b) a aceluiași pas conduce la noul flux întreg dat în fig. 6.

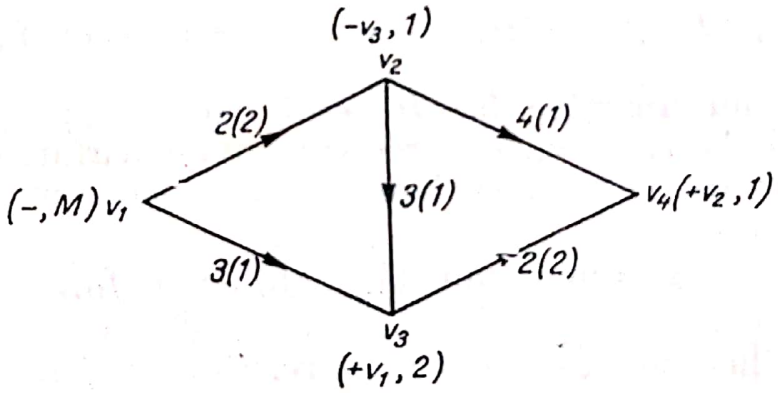


Fig. 5

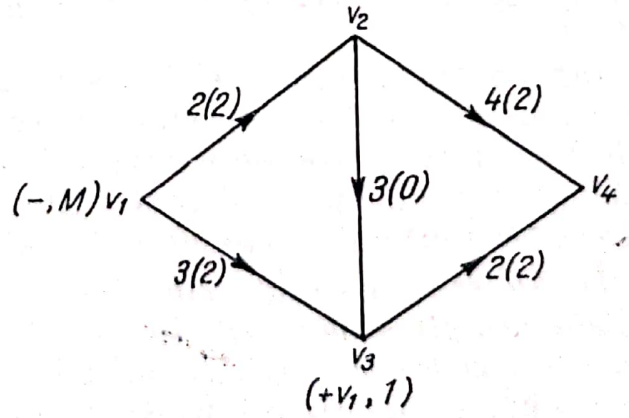


Fig. 6

Lanțul pe arcele căruia s-a modificat fluxul a fost $\mu = [v_1, v_3, v_2, v_4]$; am avut $\mu^+ = \{(v_1, v_3), (v_2, v_4)\}$, $\mu^- = \{(v_2, v_3)\}$ și $\varepsilon = 1$. Valoarea fluxului nou este mai mare cu o unitate decît valoarea fluxului inițial: $v_F = 3$, $v_{F'} = 4$. Trecem la un nou pas al algoritmului și mărcile atribuite vîrfurilor ca urmare a operației a) din acest nou pas sînt date de asemenea în ultima figură. Acum, se prezintă cazul a_2) și, prin urmare, fluxul întreg disponibil este maxim. Mulțimea vîrfurilor marcate este $H = \{v_1, v_3\}$ și ea determină secțiunea $S_H = \{(v_1, v_2), (v_3, v_4)\}$ a cărei capacitate este $k(S_H) = 4$; aceasta este secțiunea minimă. Se observă că are loc egalitatea $v_{F'} = k(S_H)$.

Să revenim acum la demonstrația teoremei 7.3. Observăm că în teoremă nu se presupune integritatea valorilor funcției de capacitate. Totuși, demonstrația, se desfășoară constructiv asemănător celei care conduce la rezolvare în algoritmul dat mai sus.

Fie H mulțimea de vîrfuri construită cu ajutorul fluxului maxim $F = (f_{ij})$, a cărui existență este presupusă în ipoteza teoremei, după procedeul următor:

— $v_1 \in H$;

— dacă $v_i \in H$ și pentru un arc (v_i, v_j) avem $f_{ij} < k_{ij}$, atunci $v_j \in H$;

— dacă $v_i \in H$ și pentru un arc (v_j, v_i) avem $f_{ji} > 0$, atunci $v_j \in H$.

Vîrful v_n nu aparține lui H, căci dacă ar aparține ar exista un lanț μ de la v_1 la v_n, cu arce care unesc vîrfuri din H și am putea modifica fluxul pe arcele lanțului ca în algoritmul precedent cu

$$\varepsilon = \min \left(\min_{\mu^+} (k_{ij} - f_{ij}), \min_{\mu^-} f_{ij} \right), \tag{7.25}$$

obținînd un flux cu valoare mai mare. Cum $v_n \notin H$, mulțimea de vîrfuri H determină o secțiune S_H cu capacitatea $k(S_H)$ și pe arcele ce leagă vîrfurile din H cu vîrfurile exterioare lui H au loc implicațiile (7.22). Pe baza egalității (7.17) rezultă $k(S_H) = v_F$ și teorema 7.3 este demonstrată.

Remarcăm că demonstrația teoremei 7.3 s-a bazat în mod esențial pe ipoteza existenței unui flux maxim. Totuși, în cazul cînd funcția de capacitate a rețelei are numai valori întregi am stabilit existența unui flux maxim în mod constructiv; mai mult, acesta este un flux întreg. Astfel, am demonstrat teorema următoare:

Teorema 7.4. *Într-o rețea redusă unde funcția de capacitate are valori întregi, există un flux maxim întreg F cu valoarea v_F și $v_F = k(S_H)$ pentru orice secțiune minimă S_H cu capacitatea $k(S_H)$.*

Remarcăm de asemenea că fluxul maxim într-o rețea redusă nu este, în general, unic și că pot exista mai multe secțiuni minime.

Scanned with OKEN Scanner

Algoritmul lui L. R. Ford și D. R. Fulkerson construiește un singur flux maxim și o singură secțiune minimă.

7.3. ALGORITMUL PRIMAL-DUAL PENTRU REZOLVAREA PROBLEMEI TRANSPORTURILOR

Problema de transport pe care o vom considera în continuare este problema de a se minimiza funcția

$$f = \sum_{i=1}^{n} \sum_{k=1}^{m} c_{ik} x_{ik} \tag{7.26}$$

cu restricțiile

$$\sum_{k=1}^{m} x_{ik} \leqslant a_i, \ (i = 1, \ldots, n), \tag{7.27}$$

$$\sum_{i=1}^{n} x_{ik} \geqslant b_k, \ (k = 1, \ldots, m), \tag{7.28}$$

și

$$x_{ik} \geqslant 0, \ (i = 1, \ldots, n; \ k = 1, \ldots, m). \tag{7.29}$$

Problema se deosebește de cea formulată în § 2.1.3 doar prin aceea că în restricții avem inecuații în loc de ecuații; dealtfel, această formă a restricțiilor exprimă condițiile ca disponibilele din depozite să nu fie depășite iar cererea să fie satisfăcută.

Remarcăm că o condiție necesară pentru ca problema să aibă soluții este $\sum_{i=1}^{n} a_i \geqslant \sum_{k=1}^{m} b_k$; într-adevăr, inegalitatea rezultă prin sumarea tuturor inecuațiilor (7.27) și apoi (7.28) urmată de compararea rezultatelor. Presupunem condiția satisfăcută și mai presupunem integritatea tuturor datelor problemei, numerele a_i, b_k și c_{ik}, $(i = 1, \ldots, n; \ k = 1, \ldots, m)$.

Problemei de transport i se poate asocia o rețea de transport în modul următor:

— fiecărui depozit D_i, $(i = 1, \ldots, n)$, și fiecărui centru de consum C_k, $(k = 1, \ldots, m)$, i se asociază cîte un vîrf al grafului,

așa încît $V = D \cup C$, unde $D = \{D_i\}$, $C = \{C_k\}$; fiecărei perechi de vîrfuri (D_i, C_k) i se asociază cîte un arc al grafului, așa încît

$$U = \{(D_i, C_k) \mid D_i \in D, C_k \in C\};$$

— funcția de cerere este definită prin

$$d(v) = \begin{cases} -a_i & \text{cînd } v \in D_i, \\ b_k & \text{cînd } v \in C_k, \end{cases} \quad (v \in V); \tag{7.30}$$

— funcția de cost este definită prin

$$c(D_i, C_k) = c_{ik}, \ (i = 1, \ldots, n; \ k = 1, \ldots, m) \tag{7.31}$$

Rețeaua nu are o funcție de capacitate.

Rețeaua asociată problemei de transport este, prin urmare, de tipul din fig. 7.

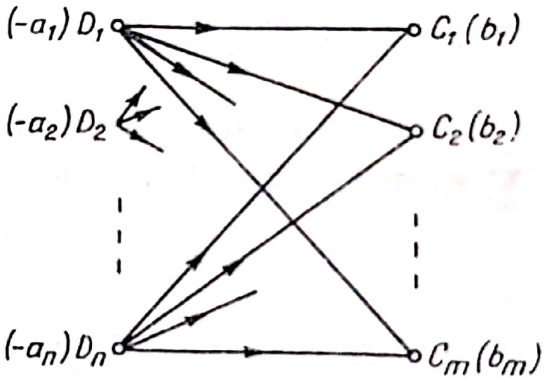


Fig. 7

Observăm că graful rețelei are un aspect particular, anume mulțimea vîrfurilor este partiționată în două submulțimi D și C, astfel încît oricare arc al grafului are extremitatea inițială în D și extremitatea finală în C; un graf cu această proprietate este denumit *graf bipartit*. Funcția de cerere determină exact aceeași partiție a vîrfurilor rețelei: D este mulțimea punctelor de disponibil și C este mulțimea punctelor de cerere; puncte de tranzit nu există. Rețeaua nu este o rețea redusă.

Pentru o rețea de acest tip este necesar să redefinim fluxul. Vom numi *flux* în rețeaua R introdusă mai înainte o funcție $F=(f(u))$, definită pe mulțimea arcelor și care satisface condiții de conservare în orice vîrf și condiții de nenegativitate pe orice arc:

$$\sum_{u \in U_v^+} f(u) - \sum_{u \in U_v^-} f(u) \leqslant -d(v), \quad (v \in V), \tag{7.32}$$

$$f(u) \geqslant 0, \ (u \in U), \tag{7.33}$$

Scanned with OKEN Scanner

unde

$$U_v^+ = \{u \mid u = (v, w),\ u \in U\},$$
$$U_v^- = \{u \mid u = (w, v),\ u \in U\}. \tag{7.34}$$

Remarcăm sensul practic al condițiilor de conservare (7.32): fluxul net ce iese dintr-un punct de disponibil nu depășește disponibilul și fluxul net ce intră într-un punct de cerere satisface cererea.

Dacă introducem notația cu doi indici pentru flux și scriem condițiile de conservare pe rînd pentru toate vîrfurile rețelei, ținînd seamă și de valorile funcției de cerere, obținem exact restricțiile (7.27), (7.28); condițiile (7.33) se transformă în condițiile (7.29).

Pentru fiecare flux în rețea, $F = (f(u))$, este definit costul total $\sum_{u \in U} f(u)\, c(u)$ sau în notația cu indici expresia (7.26). Atunci, în limbajul teoriei fluxurilor în rețele, problema transporturilor este problema de a se determina în mulțimea tuturor fluxurilor fluxul de cost total minim. Este clar că problema transporturilor reprezintă un caz particular de problemă de flux cu cost minim, extensiunile problemei transporturilor admițînd rețele cu grafuri de orice tip, cu o funcție de cerere permițînd existența punctelor de tranzit, cu funcție de capacitate, cu funcție de cost mai generală etc. În continuare, vom prezenta algoritmul primal-dual pentru rezolvarea problemei de transport, deși un algoritm asemănător se poate formula pentru rezolvarea unei probleme de flux de cost minim mult mai generale.

Algoritmul rezolvă în același timp problema de transport și duala sa; de aici provine denumirea de algoritm primal-dual. Acum apare justificată scrierea problemei de transport cu restricții inecuații. Într-adevăr, problema împreună cu duala sa vor reprezenta o pereche de probleme duale simetrice. Ținînd seamă că restricțiile problemei de transport se scriu sub forma

$$-\sum_{k=1}^{m} x_{ik} \geqslant -a_i, \quad (i = 1, \ldots, n),$$
$$\sum_{i=1}^{n} x_{ik} \geqslant b_k, \quad (k = 1, \ldots, m), \tag{7.35}$$

Scanned with OKEN Scanner

și notînd variabilele duale corespunzătoare cu u_i și v_k, $(i = 1, \ldots, n;\ k = 1, \ldots, m)$, vom scrie problema dual sub forma: să se maximizeze

$$g = -\sum_{i=1}^{n} a_i u_i + \sum_{k=1}^{m} b_k v_k, \tag{7.36}$$

cu restricțiile

$$-u_i + v_k \leqslant c_{ik}, \quad (i = 1, \ldots, n;\ k = 1, \ldots, m), \tag{7.37}$$

și

$$u_i \geqslant 0,\ v_k \geqslant 0, \quad (i = 1, \ldots, n;\ k = 1, \ldots, m). \tag{7.38}$$

Problema dual a fost scrisă conform regulilor date în § 5.1 Condițiile de complementarietate se scriu

$$x_{ik}(c_{ik} + u_i - v_k) = 0, \tag{7.39}$$

$$\left(a_i - \sum_{k=1}^{m} x_{ik}\right) u_i = 0, \tag{7.40}$$

$$\left(\sum_{i=1}^{n} x_{ik} - b_k\right) v_k = 0, \tag{7.41}$$

$$(i = 1, \ldots, m;\ k = 1, \ldots, n).$$

Algoritmul se bazează pe teorema 5.6 care se exprimă astfel: $\{x_{ik}\}$ și $\{u_i, v_k\}$ sînt soluții optime ale perechii de probleme duale dacă și numai dacă $\{x_{ik}\}$ satisfac condițiile de admisibilitate primală (7.27), (7.28), (7.29), $\{u_i, v_k\}$ satisfac condițiile de admisibilitate dual (7.37), (7.38) și împreună satisfac condițiile de complementarietate (7.39), (7.40), (7.41).

Ideea central a algoritmului este următoarea: în fiecare pas dispunem de un sistem de valori ale variabilelor primale, de o soluție admisibil a problemei duale și cele două sisteme de valori ale variabilelor satisfac împreună condițiile de complementarietate (7.39), (7.40); dacă în cursul unui pas s-a obținut un sistem de valori ale variabilelor primale care este o soluție admisibil a problemei de transport, construită astfel ca să satisfacă condițiile de complementarietate împreună cu soluția admisibil disponibil a

Scanned with OKEN Scanner

dualei, atunci s-a obţinut o pereche de soluţii optime a celor două probleme duale; în caz contrar, se modifică soluţia admisibilă a dualei astfel ca să obţinem o nouă soluţie admisibilă a dualei, care împreună cu noul sistem de valori ale variabilelor primale să satisfacă condiţiile de complementarietate (7.39), (7.40) şi se trece la un nou pas. Din descrierea principială a algoritmului rezultă că în cursul unui pas se realizează două operaţii mari: a) schimbarea sistemului de valori ale variabilelor primale; b) dacă este cazul, schimbarea soluţiei admisibile a problemei duale.

Iniţial, se poate lua

$$u_i = 0,\ v_k = \min_i c_{ik},\ x_{ik} = 0,$$
$$(i = 1, \ldots, n;\ k = 1, \ldots, m); \tag{7.42}$$

se poate verifica imediat că $\{u_i, v_k\}$ satisfac condiţiile (7.37), (7.38) şi împreună cu $\{x_{ik}\}$ condiţiile de complementarietate (7.39), (7.40). Se înţelege, se poate pleca şi de la oricare altă pereche de sisteme de valori ale variabilelor, care satisfac aceste condiţii.

În cursul oricărui pas „progresul" constă în satisfacerea încă a unei noi condiţii (7.41), pe lîngă cele care sînt deja satisfăcute. Totodată sistemul de valori ale variabilelor primale satisface în fiecare pas condiţiile de admisibilitate (7.27) şi în cursul pasului se va „progresa" prin satisfacerea unei noi condiţii (7.28).

Să trecem la descrierea operaţiilor a) şi b).

a) Se rezolvă o problemă de flux maxim într-o reţea redusă auxiliară. Anume, completăm graful reţelei asociate problemei de transport cu un vîrf-intrare fictiv D_0 şi un vîrf-ieşire fictiv C_0 şi cu arcele fictive (D_0, D_i), $(i = 1, \ldots, n)$, şi (C_k, C_0), $(k = 1, \ldots, m)$. Definim funcţia de capacitate pe arcele fictive $k(D_0, D_i) = a_i$, $(i = 1, \ldots, n)$, $k(C_k, C_0) = b_k$, $(k = 1, \ldots, m)$, apoi $k(D_i, C_k) = 0$ dacă $c_{ik} + u_i - v_k > 0$ şi pe celelalte arce ale reţelei iniţiale considerăm capacitatea oricît de mare. Rezolvarea problemei de flux începe cu fluxul obţinut la sfîrşitul pasului precedent, sau iniţial cu fluxul nul. Dacă fluxul maxim în reţeaua auxiliară saturează arcele fictive (C_k, C_0), $(k = 1, \ldots, m)$, atît condiţiile (7.28) cît şi condiţiile (7.41) sînt satisfăcute şi algoritmul este oprit; dispunem de o pereche de soluţii optime pentru problema de transport şi duala sa. În caz contrar, se trece la operaţia b).

b) Odată cu rezolvarea problemei fluxului maxim în reţeaua auxiliară, se determină mulţimea vîrfurilor marcate H. Fie

Scanned with OKEN Scanner

$I = H \cap D$, $K = H \cap C$ și $\bar{I} = D \setminus I$, $\bar{K} = C \setminus K$; atunci se modifică valorile variabilelor duale astfel:

$$u_i' = \begin{cases} u_i & D_i \in I, \\ u_i + \delta, & D_i \in \bar{I}, \end{cases} \quad (i = 1, \ldots, n), \tag{7.43}$$

$$v_k' = \begin{cases} v_k, & C_k \in K, \\ v_k + \delta, & C_k \in \bar{K}, \end{cases} \quad (k = 1, \ldots, m), \tag{7.44}$$

unde

$$\delta = \min\ (c_{ik} + u_i - v_k), \tag{7.45}$$

minimul fiind luat pe toate arcele cu extremitatea inițială în I și extremitatea finală în $\bar{K}$. Această modificare produce o modificare a rețelei auxiliare, dar s-au eliminat din rețeaua folosită în cursul pasului numai arce pe care fluxul maxim avea valori nule, așa încît în pasul următor dispunem în noua rețea de un flux de plecare, tocmai fluxul maxim de la sfîrșitul pasului precedent.

Se demonstrează că algoritmul primal-dual schițat este convergent (v. [13], § 3.1). Subliniem că algoritmul prezentat este numai unul dintre algoritmii folosiți în mod curent pentru rezolvarea problemelor de transport. Există mulți algoritmi cu rezultate numerice deosebite pentru rezolvarea aceleiași probleme și încă mai mulți pentru rezolvarea unor probleme de transport particulare.

Comentarii asupra bibliografiei

Bibliografia cuprinde mai multe categorii de lucrări: cele indicate cu [4 — 10] au fost introduse pentru a se arăta unde au fost publicate pentru prima oară metodele dezvoltate în cele șapte capitole ale lucrării; cele indicate cu [1, 2, 11 — 13] sînt manuale sau tratate de largă circulație în literatura consacrată programării liniare; cele indicate cu [14 — 24] sînt lucrările apărute în literatura românească a domeniului; lucrarea [3] cuprinde și aplicații ale programării liniare în teoria economică.

După cum se observă cu ușurință, urmărindu-se anii de apariție a lucrărilor, perioada 1958 — 1963 este perioada apariției tratatelor de teorie generală a programării liniare, perioada 1964 — 1968 este perioada apariției primelor lucrări românești în domeniu. În ultimii zece ani aproape toate țările au avut o perioadă analoagă de elaborare a unei literaturi proprii de programare liniară, de aceea numărul cărților de programare liniară publicate în diverse limbi este foarte mare. În același timp, în literatura mondială au apărut lucrări legate de cele de teorie generală a programării liniare, anume se pare că preferințele au evoluat cel puțin în trei direcții: lucrări teoretice consacrate expunerii unor metode de rezolvare pentru programe neliniare, lucrări teoretice care dezvoltă teoria unor clase de programe liniare, lucrări cuprinzînd aplicații în

Scanned with OKEN Scanner

diverse ramuri ale economiei. Astăzi, încercarea inițială a lui V. Riley și S. Gass de a întocmi o bibliografie mondială a programării liniare a fost abandonată. În schimb, numărul proiectelor de anvergură cuprinzînd aplicații ale programării liniare este în creștere.

BIBLIOGRAFIE

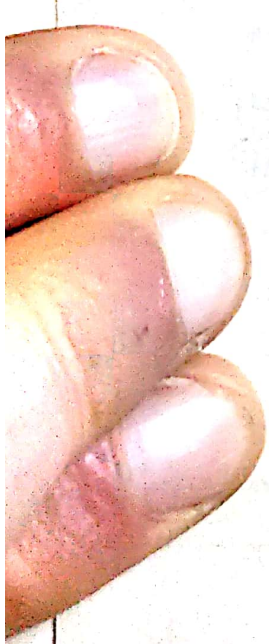
1. Gass S. *Linear programming: methods and applications.* New York, McGraw-Hill 1958.
2. Simmonard M., *Programmation linéaire.* Paris,, Dunod, 1962.
3. Nikaido, H. *Convex structure and economic theory.* New York, Acad. Press, 1968.
4. Dantzig, G. *Activity analysis of production and allocation,* (ed. T. Koopmans). New York, J. Wiley, 1951.
5. Lemke, C. *The dual method of solving the linear programming problem.* Technicaf Report 29, Carnegie Institute of Technology, 1953.
6. Gass, S., Saaty T. *The computational algorithm for the parametric objective function.* NRL Q, 2, 1—2, 1955.
7. Gass, S., Saaty, T. *Parametric objective function — Part II: generalization.* Op. Res., 3, 4, 1955.
8. Gomory, R. *Outline of an algorithm for integer solutions to linear programs.* Bull. Amer. Math. Soc., 64, 5, 1958.
9. Gomory R. *An algorithm for integer solutions to linear programs.* În Recent advances in mathematical programming, (ed. R. Graves, Ph. Wolfe). New York, McGraw Hill, 1963.
10. Korte B. *Ein lexicographischer Suchalgorithmus zur Lösung allgemeiner ganzzahliger Programmierungsaufgaben.* Unternehmensforschung, 13, 2, 1969.
11. Hadley, G. *Linear programming.* Addison-Wesley, Reading Massachusets, 1962.
12. Dantzig, G. *Linear programming and extensions.* Princeton University Press, 1963.
13. Ford, L. R., Fulkerson, D. R. *Flows in networks.* Princeton University Press, 1962.
14. Dragomirescu, M., Malița M. *Programarea pătratică. Introducere în programarea convexă.* București, Editura științifică 1968.
15. Dragomirescu, M., Malița, M. *Programarea neliniară.* București, Editura științifică, 1972.
16. Ionescu H., Mirescu P., Năndejde, I. *Programarea liniară.* București, Editura științifică, 1964.
17. Lupescu, T., Roșu, A., Cerchez, M. *Programare matematică.* București, Editura militară, 1965.
18. Malița, M., Zidăroiu, C., *Matematica organizării.* București, Editura tehnică 1972.
19. Marușciac, I. *Metode de rezolvare a problemelor de programare neliniară.* Cluj, Editura Dacia, 1973.
20. Mihoc, Gh., Ionescu, H. *Bazele matematice ale programării liniare.* București, Editura științifică, 1965.
21. Mihoc, Gh., Năndejde, I. *Programarea matematică.* Vol. I—II, București, Editura științifică, 1966—1968.
22. Mihoc, Gh., Stefănescu, A. *Programarea matematică.* București, Editura didactică și pedagogică, 1973.
23. Mihăilă, N. *Introducere în programarea liniară.* București, Editura didactică și pedagogică, 1970.
24. Tamaș, V. *Lecții de programare matematică.* Universitatea Iași, 1975.

Scanned with OKEN Scanner

BASIC TECHNIQUES IN LINEAR PROGRAMMING

Linear programming is one of the mathematical fields, which offers a theoretical basis for operations research. The general linear programming problem, that of maximizing or minimizing a linear function subject to linear constraints and non negativity conditions generated a group of techniques and many other related optimization problems.

The present work tries to give the well-known techniques in linear programming and the most important related problems, such that the reader becomes prepared from theoretical point of vue for using a computer. The primal and the dual simplex method are given in the first part; the second one is devoted to related optimization problems as parametric programming, integer programming and transportation problem. The book can be used like a textbook for a one semester course in linear programming, for undergraduate students in economics, engineering or mathematics.



Scanned with OKEN Scanner